

## 功能特色：

\*内置**256x8 EEPROM** (无需I2C协议，寄存器控制擦写)

\***LVD**复用**LVR**（通过软件控制）

\*四路独立控制定时器（**T0\TC0\TC1\TC2**）

\*三路独立控制**PWM/BUZZER**输出（**PB2\PB1\PB0**）

\*硬件控制端口强弱驱动电流

\***PB3**口可做正常**IO**输出高低

## 注意事项：

- 1、 B3 口上拉问题：打开 B3 口上拉后，接低电平会导致芯片复位，不要使用 B3 口上拉功能。
- 2、 外置晶振问题：外置晶振不能和内置 EEPROM 同时使用，否则程序会出现错乱现象。
- 3、 双时钟问题：双时钟定时误差较大，不建议使用。
- 4、 配置区：LVDT 必须选择 Disable，看门狗必须选择打开。否则芯片在上电复位时有概率出现配置位载错问题。若方案中必须需要用到 LVDT，则建议选择 2.6V 及以上，并进行详细验证。

## 修正记录

版本号	日期	内容
V10	2016 年 6 月	初版
V20	2017 年 7 月	更新芯片电压频率关系，增加SOP8封装脚位，增加注意事项
V30	2017 年 10 月	增加封装信息
V40	2017 年 12 月	增加注意事项

## OTP-Based 8-Bit Microcontroller Series

- SM701 : OTP device

### 功能特性

- 只有42个单字指令
- 除跳转指令为两个周期指令以外其余为单周期指令
- 13-bit指令宽度
- GOTO指令能跳转到所有的ROM/EPROM地址空间
- 子程序能返回到所有的ROM/EPROM地址空间
- 能处理8位数据
- 5级硬件堆栈
- 运行速度: DC-20 MHz 工作频率  
DC-100 ns 指令周期

型号	管脚#	I/O#	EPROM/ROM空间 (Byte)	RAM (Byte)
SM701	14	12	1K	48

- 支持直接与间接数据寻址方式
- 一个带8位预置器的8位定时/计数器 (Timer0)，三个带8位自动重载功能的8位定时/计数器 (TC0, TC1, TC2)
- 内部上电复位
- 内含一个低电压检测电路供掉电复位使用
- 上电复位计数器 (PWRT) 和振荡启动计数器 (Oscillator Start-up Timer OST)
- 内部振荡器集成了一个看门狗保证了可靠的操作同时软件使能看门狗操作
- 两类双向输入输出I/O口IOA 和 IOB
- 三路独立控制可编程PWM/蜂鸣器输出
- 通过编程控制I/O端口的上拉/下拉、开漏等状态
- 内置256byte可擦写EEPROM/ (共16个sector, 每个sector含有8个byte)
- 四个内部计数中断源; 两个外部中断源: INT管脚, PortB的输入改变
- 通过INT管脚或者PortB的输入改变来实现睡眠唤醒
- 省电睡眠模式
- 内部有8MHz, 4MHz, 1MHz, 和 455KHz RC振荡器
- 有可靠的保证使得程序代码不被读出。
- 内部RC振荡器
- 提供以下振荡源的选择:
  - ERC: External Resistor/Capacitor Oscillator (外部的RC振荡器)
  - IRC/ERIC: Internal or External Resistor/Internal Capacitor Oscillator – (内部电阻内部的电容RC振荡器或外部的电阻内部的电容RC振荡器)
  - IRC\_RTC: Internal or External Resistor/ Low Frequency Crystal Oscillator For T0 RTC
  - HF: High Frequency Crystal/Resonator Oscillator (高频率的晶体振荡器)
  - LF: Low Frequency Crystal Oscillator (低频率的晶体振荡器)
  - XT: Crystal/Resonator Oscillator (晶体/陶瓷振荡器)
- 工作电压范围:2.0V - 5.5V
  - ≤1MHZ: 2.0V - 5.5V
  - ≤2MHZ: 2.2V - 5.5V
  - ≤4MHZ: 2.6V - 5.5V

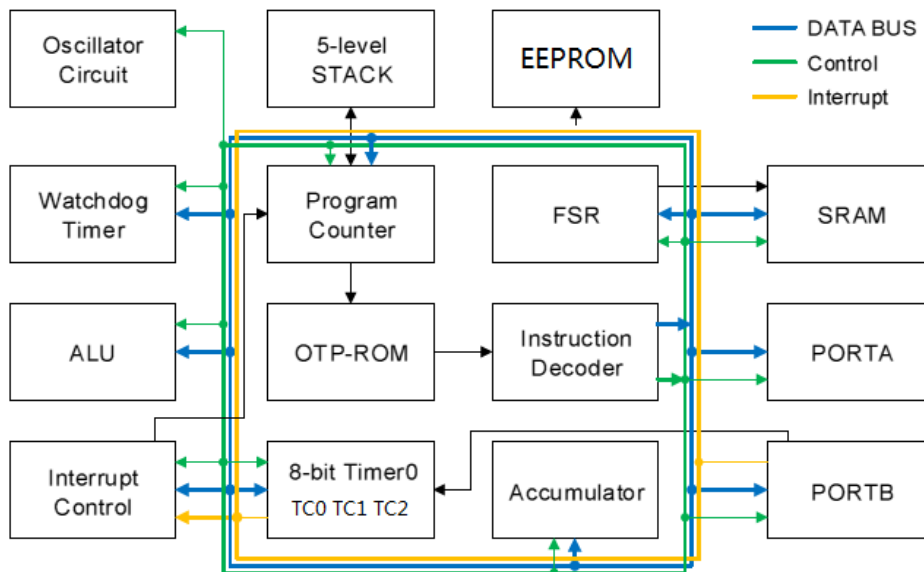
SM701是一款低功耗，高速，高噪声容限，EPROM/ROM基于8位CMOS工艺制造的单片机，采用RISC指令集，共有42条指令，除分支指令为两个周期指令以外其余为单周期指令。这种易用、易记的指令集大大缩短了开发时间。

SM701包含了上电复位(Power-on Reset POR)，掉电复位(Brown-out Reset BOR)，上电复位计数器 (Power-up Reset Timer PWRT)，振荡启动计数器 (Oscillator Start-up Timer OST)，看门狗定时器(Watchdog Timer)，EPROM/ROM, EEPROM, SRAM, 双向三态I/O口，(可以设置为上拉/下拉、开漏)，省电睡眠模式，一个带8位预置器的8位定时/计数器，三个带8位自动重载的8位定时器/计数器，独立中断，睡眠唤醒模式和可靠的代码保护，有两个振荡源可供用户配置选择，包含省电振荡源和低功耗振荡器。

SM701可访问1K×13的程序存储空间。

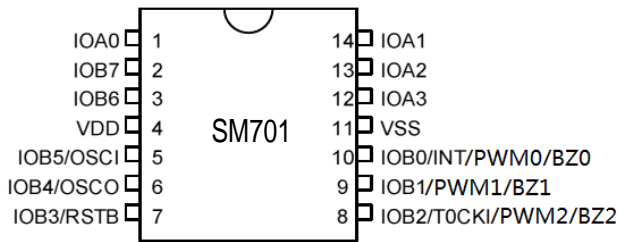
SM701能直接或间接访问寄存器以及数据存储区，所有的特殊功能寄存器分布在数据存储区同时包含特定的程序指针。

## 结构图

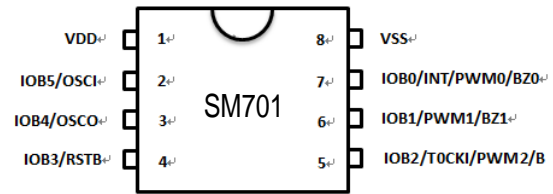


## 管脚图

PDIP14, SOP14



PDIP8, SOP8



## 管脚功能描述

管脚名称	I/O	说明
IOA0 ~ IOA3	I/O	IOA0 ~ IOA3 双向I/O口 软件可以设置为下拉
IOB0/INT/PWM0	I/O	双向I/O口同时具有系统唤醒功能 软件设置为上拉/下拉和开漏 外部中断输入脚 <a href="#">Buzzer0输出引脚/PWM0输出引脚</a>
IOB1/PWM1	I/O	双向I/O口同时具有系统唤醒功能 软件设置为上拉/下拉和开漏 <a href="#">Buzzer1输出引脚/PWM1输出引脚</a>
IOB2/T0CKI/PWM2	I/O	双向I/O口同时具有系统唤醒功能 软件设置为上拉/下拉和开漏 外部计数输入脚 <a href="#">Buzzer2输出引脚/PWM2输出引脚</a>
IOB3/RSTB	I/O	<a href="#">双向I/O同时具有系统唤醒功能</a> <a href="#">软件设置为上拉/硬件设置开漏</a> 系统复位脚。低电平复位。设置为复位脚时上拉自动开启
IOB4/OSCO	I/O	双向I/O口同时具有系统唤醒功能 (RCOUT 可选择IRC/ERIC, ERC模式) 软件设置为上拉/开漏 晶体振荡器输出脚 (HF, LF, IRC_RTC模式) 基于指令周期晶体振荡器输出 (RCOUT 可选择IRC/ERIC, ERC模式)
IOB5/OSCI	I/O	双向I/O口同时具有系统唤醒功能 (IRC 模式) 软件设置为上拉/开漏 晶体振荡器输入脚 (HF, LF, IRC_RTC模式) 外部实时时钟输入脚(ERIC, ERC模式)
IOB6 ~ IOB7	I/O	双向I/O口同时具有系统唤醒功能 软件设置为上拉/开漏
Vdd	-	电源
Vss	-	地

Legend: I=输入, O=输出, I/O=输入/输出

## 1.0 存储器结构

SM701 存储器包含程序存储器和数据存储器。

### 1.1 程序存储器

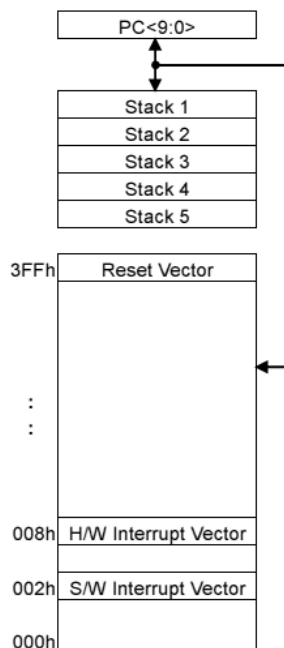
SM701有一个10位PC指针能访问1Kx13的存储空间。

SM701的复位地址为3FFh。

H/W中断向量地址008h., S/W中断向量地址002h。

SM701的CALL/GOTO能指向在同一个程序页面（一个程序页面为1K）的所有存储空间

### 程序存储器分布图和堆栈结构



SM701

### 1.2数据存储器

数据存储器包含特殊功能器组和通用寄存器组，所有通用寄存器可以直接寻址或者通过FSR寄存器间接寻址。

特殊功能寄存器用来控制CPU或外围功能模块的工作。

表 1.1: SM701寄存器列表

Address	Description Bank0	Description Bank1	Description Bank2
00h	INDF		
01h	TMR0		
02h	PCL		
03h	STATUS		
04h	FSR		
05h	PORTA		
06h	PORTB		
07h	TCXCON	E2PCON	TC0M
08h	PCON	E2PADR	TC0C
09h	WUCON	E2PDIN	TC0R
0Ah	PCHBUF	E2PDATA	TC1M
0Bh	PDCON		TC1C
0Ch	ODCON		TC1R
0Dh	PHCON		TC2M
0Eh	INTEN		TC2C
0Fh	INTFLAG		TC2R
10h ~ 3Fh	General Purpose Registers		

N/A  OPTION

05h  IOSTA

06h  IOSTB

表1.2: 通过OPTION 或IOST指令控制的寄存器

地址	说明	B7	B6	B5	B4	B3	B2	B1	B0
N/A (w)	OPTION	TOTB	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
05h (w)	IOSTA	Port A I/O控制寄存器							
06h (w)	IOSTB	Port B I/O控制寄存器							

表1.3: 寄存器列表

地址	说明	B7	B6	B5	B4	B3	B2	B1	B0
00h (r/w)	INDF	通过FSR 访问数据区(不是一个实际的物理地址)							
01h (r/w)	TMR0	8位定时/计数器							
02h (r/w)	PCL	低8位PC指针							
03h (r/w)	STATUS	RST	GP1	GP0	/TO	/PD	Z	DC	C
04h (r/w)	FSR	RP1	RP0	间接地址访问指针 (RAM选择寄存器)					
05h (r/w)	PORTA	-	-	-	-	IOA3	IOA2	IOA1	IOA0
06h (r/w)	PORTB	IOB7	IOB6	IOB5	IOB4	IOB3	IOB2	IOB1	IOB0

地址Bank0	说明	B7	B6	B5	B4	B3	B2	B1	B0
07h (r/w)	TCXCON	NG	NG	NG	NG	TC10EN	PWM1INV	TC20EN	PWM2INV
08h (r/w)	PCON	WDTE	EIS	LVDTE	LVDF	LVDRB	LVDSSEL2	LVDSSEL1	LVDSSEL0
09h (r/w)	WUCON	WUB7	WUB6	WUB5	WUB4	WUB3	WUB2	WUB1	WUB0
0Ah (r/w)	PCHBUF	-	-	-	-	-	-	2 MSBs Buffer of PC	
0Bh (r/w)	PDCON	*	/PDB2	/PDB1	/PDB0	/PDA3	/PDA2	/PDA1	/PDA0
0Ch (r/w)	ODCON	ODB7	ODB6	ODB5	ODB4	-	ODB2	ODB1	ODB0
0Dh (r/w)	PHCON	/PHB7	/PHB6	/PHB5	/PHB4	/PHB3	/PHB2	/PHB1	/PHB0
0Eh (r/w)	INTEN	GIE	*	TC2IE	TC1IE	TC0IE	INTIE	PBIE	TOIE
0Fh (r/w)	INTFLAG	-	-	TC2IF	TC1IF	TC0IF	INTIF	PBIF	TOIF

地址Bank1	说明	B7	B6	B5	B4	B3	B2	B1	B0
07h (r/w)	E2PCON	-	-	-	-	E2PCER	E2PSER	E2PBER	E2PWR
08h (r/w)	E2PADR	E2PADR7	E2PADR6	E2PADR5	E2PADR4	E2PADR3	E2PADR2	E2PADR1	E2PADR0
09h (r/w)	E2PDIN	E2PDIN7	E2PDIN6	E2PDIN5	E2PDIN4	E2PDIN3	E2PDIN2	E2PDIN1	E2PDIN0
0Ah (r)	E2PDATA	E2PDA7	E2PDA6	E2PDA5	E2PDA4	E2PDA3	E2PDA2	E2PDA1	E2PDA0

地址Bank2	说明	B7	B6	B5	B4	B3	B2	B1	B0
07h (r/w)	TC0M	TC0EN	TC0PS2	TC0PS1	TC0PS0	TC0CKS-	ALOAD0	TC0OUT	PWM0EN
08h (r/w)	TC0C	TC0C7	TC0C6	TC0C5	TC0C4	TC0C3	TC0C2	TC0C1	TC0C0
09h (r/w)	TC0R	TC0R7	TC0R6	TC0R5	TC0R4	TC0R3	TC0R2	TC0R1	TC0R0
0Ah (r/w)	TC1M	TC1EN	TC1PS2	TC1PS1	TC1PS0	TC1CKS	ALOAD1	TC1OUT	PWM1EN
0Bh (r/w)	TC1C	TC1C7	TC1C6	TC1C5	TC1C4	TC1C3	TC1C2	TC1C1	TC1C0
0Ch (r/w)	TC1R	TC1R7	TC1R6	TC1R5	TC1R4	TC1R3	TC1R2	TC1R1	TC1R0
0Dh (r/w)	TC2M	TC2EN	TC2PS2	TC2PS1	TC2PS0	TC2CKS	ALOAD2	TC2OUT	PWM2EN
0Eh (r/w)	TC2C	TC2C7	TC2C6	TC2C5	TC2C4	TC2C3	TC2C2	TC2C1	TC2C0
0Fh (r/w)	TC2R	TC2R7	TC2R6	TC2R5	TC2R4	TC2R3	TC2R2	TC2R1	TC2R0

Legend: - = unimplemented, read as '0', \* = unimplemented, read as '1', NG= no used bit

## 2.0 功能介绍

### 2.1 寄存器操作

#### 2.1.1 INDF (间接寻址寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
00h (r/w)	INDF	通过FSR 访问数据区(不是一个实际的物理地址)							

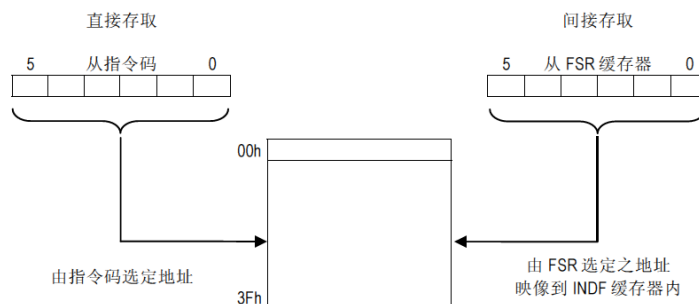
INDF不是一个实际的物理地址，间接寻址时INDF通过RAM选择寄存器（FSR）来访问其所指向的地址。间接寻址读操作直接读地址00h(FSR="0")，间接寻址不能对INDF直接进行写操作（尽管有些状态会发生改变）。

FSR的5-0位可以用来选择64个寄存器（地址:00h ~ 3Fh）。

#### 例 2.1:间接寻址

- 地址38内容为10h
- 地址39内容为0Ah
- 将38写入FSR 中
- 通过A读INDF返回10h
- FSR加1 (@FSR=39h)
- 通过A读INDF返回0A h

图2.1:直接/间接存取



#### 2.1.2 TMR0 (定时/计数器 Time lock/Counter register)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
01h (r/w)	TMR0	8位定时/计数器							

TMR0是一个8位定时/计数器寄存器，Timer0的时钟源可以取值于指令周期或外部实时钟（T0CKI pin），使用外部时钟需要设置OPTION的T0CS(T0CS=5)位为1。

使用TMR0的预置器需要设置OPTION的PSA (PSA =3)位为0，这种模式下TMR0值的改变，预置器被清零。

#### 2.1.3 PCL (Low Bytes of Program Counter) & Stack

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
02h (r/w)	PCL	PC低8位							

SM701的PC指针和堆栈的位数为10位，堆栈有5级，低位的PC指针为PCL寄存器，该寄存器时可读写的，高位的PC指针为PCH寄存器，该寄存器包含PC<9:8>位，该寄存器不能直接读写。PCH寄存器的改变是通过PCHBUF寄存器来实现的。每一条指令执行的时候他的PC指针包含下一条指令的操作地址。指令没有改变PC内容时候、在每一个指令周期PC指针自动加1。

对于GOTO指令有PC<9:0>，PCL 映射成PC<7:0>，PCHBUF不变。

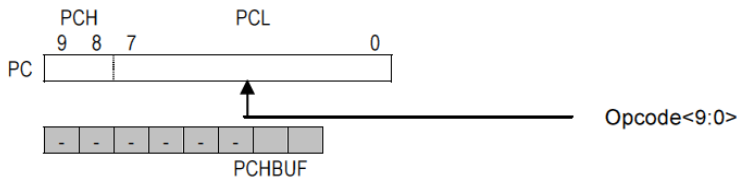
对于CALL指令有PC<9:0>，下一条指令地址被推进堆栈，PCL 映射成PC<7:0>，PCHBUF不变。

对于RETI, RETFIE, RETURN指令有PC<9:0>，PC的内容更改为出栈信息，PCL 映射成PC<7:0>，PCHBUF不变。

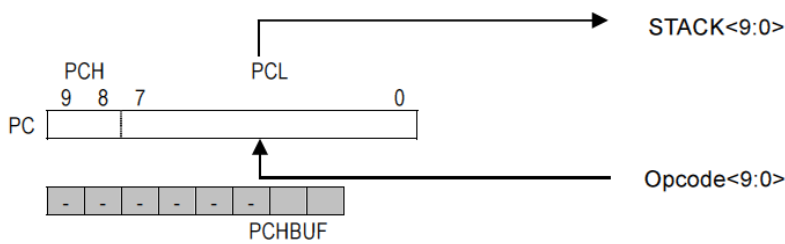
对于其他指令，PCL就是目标信息，PC<7:0>的内容就是指令地址或。不管怎样，PC<9:8>来源于PCHBUF<1:0>位 (PCHBUF→PCH)。PCHBUF不会改变，从而PCH不会改变。

图2.2:不同的指令调用PC指针跳转方式

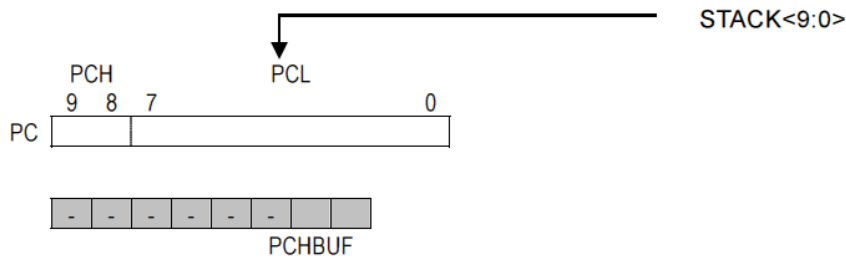
### 1、GOTO指令



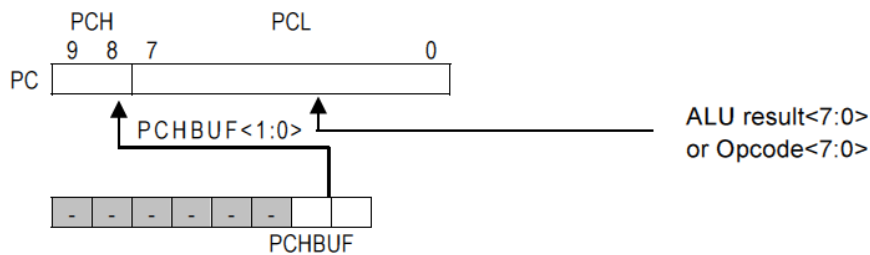
### 2、CALL指令



### 3、RETIA, RETFIE, RETURN指令



### 4、以PCL为目的的指令



注释1. PCHBUF只有在PCL内容是目标地址才有效，当PCL是运算结果时候，PCHBUF不起作用。



### 2.1.4 STATUS (状态字寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
03h (r/w)	STATUS	RST	GP1	GP0	/TO	/PD	Z	DC	C

状态字寄存器包含运算标志，结果标志。

指令执行以后可能会影响STATUS寄存器的Z、DC、C标志位，则不能直接对这三个标志位进行写操作，这些标志位的设置由MCU的逻辑自动完成。同时，TO和PD位也是不能通过指令直接改变写操作。因此，与STATUS作为目标寄存器的指令后，结果可能会与预期的不同。例如：运行CLRSTATUS将把STATUS的高三位置零和Z标志位置1同时该寄存器的内容如下

0	0	0	u	u	1	u	u
---	---	---	---	---	---	---	---

u表示为指令执行前后该位没有发生改变

#### C:进位标志

ADDAR, ADDIA

= 1, 有进位

= 0, 无进位

SUBAR, SUBIA

= 1, 无借位

= 0, 有借位

注释：减法是通过将2的补第二个操作数的执行。旋转(RRR, RLR)指令，该位装载高或低位源寄存器位。

#### DC:辅助进位/借位标志.(低四位向高四位进位/借位标志)

ADDAR, ADDIA

= 1, 底4位有进位

= 0, 底4位无进位

SUBAR, SUBIA

= 1, 底4位无借位

= 0, 底4位有借位

#### Z:零标志位

= 1, 算术或逻辑运算结果为“0”时

= 0, 算术或逻辑运算结果不为“0”时

#### /PD:系统休眠标志位

= 1, 当系统上电时或执行“CLRWDT”指令后

= 0, 当执行“SLEEP”指令后

#### /TO:看门狗溢出标志位

= 1, 当系统上电时或执行“CLRWDT”或SLEEP指令后

= 0, 看门狗定时器溢出

#### GP1:GP0:通用寄存器读/写位

#### RST:定义系统复位类型位.

= 1, 唤醒SLEEP或Port B脚位变化唤醒SLEEP

= 0, 其他类型唤醒SLEEP.

### 2.1.5 FSR (间接寻址指针)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
04h (r/w)	FSR	RP1	RP0	间接寻址指针					

**Bit5:Bit0** : 用来选择访问间接寻址时目标寄存器地址. 具体描述见2.1.1。

**Bit7:Bit6** : 用来选择控制寄存器的页，控制寄存器页选择只对地址07H~0FH有效，地址00H~06H和控制寄存器页选择无关。

= 00, 选择bank0

= 01, 选择bank1

= 10, 选择bank2

= 11, 没有使用

## 2.1.6 PORTA, PORTB (Port 寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
05h (r/w)	PORTA	-	-	-	-	IOA3	IOA2	IOA1	IOA0
06h (r/w)	PORTB	IOB7	IOB6	IOB5	IOB4	IOB3	IOB2	IOB1	IOB0

读端口(PORTA, PORTB 寄存器)的状态依赖于该端口是输入/输出模式, 写端口是向锁存器写数据。  
PORTA 是一个4位端口数据寄存器, 只有低4位被使用 (PORTA<3:0>). Bit7~Bit4没有使用, 置0。  
PORTB是一个8位端口数据寄存器。

## 2.1.7 TCXCON (TCX定时器模式控制寄存器)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
07h (r/w)	TCXCON	NG	NG	NG	NG	TC10EN	PWM1INV	TC20EN	PWM2INV

**PWM2INV:** IOB2口输出PWM或者BUZZER波形取反  
= 0, IOB2口输出PWM2或BUZZER2波形  
= 1, IOB2口输出PWM2反向或者BUZZER2反向波形

**TC20EN:** TC2定时器使能信号选择  
= 0, TC2定时器使能信号为TC2EN  
= 1, TC2定时器使能信号为TC0EN

**PWM1INV:** IOB1口输出PWM或者BUZZER波形取反  
= 0, IOB1口输出PWM1或BUZZER1波形  
= 1, IOB1口输出PWM1反向或者BUZZER1反向波形

**TC10EN:** TC1定时器使能信号选择  
= 0, TC1定时器使能信号为TC1EN  
= 1, TC1定时器使能信号为TC0EN

**Bit7:Bit4:** 通用读写位

## 2.1.8 PCON (电源控制寄存器)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
08h (r/w)	PCON	WDTE	EIS	LVDTE	LVDF	LVDRB	LVDSEL2	LVDSEL1	LVDSEL0

**LVDSEL2:LVDSEL0:** 当然LVDRB=1时, 选择LVD检测点

- = 000, NO LVD
- = 001, LVD=2.0V, 睡眠时关闭
- = 010, LVD=2.0V
- = 011, LVD=3.6V
- = 100, LVD=1.8V
- = 101, LVD=2.2V
- = 110, LVD=2.4V
- = 111, LVD=2.6V

**LVDRB :** LVD功能复用选择位

- = 0, LVD作为低压复位点
- = 1, LVD作为低压检测点

**LVDF :** LVD低电压检测标志位

- = 0, VDD高于设定的低压检测点
- = 1, VDD低于设定的低压检测点

**LVDTE :** LVDT (低电压检测) 使能位

- = 0, 关闭 LVDT
- = 1, 使能 LVDT

**EIS :** 定义管脚B0/INT功能位

- = 0, IOB0 (双向I/O 口) is selected. 屏蔽了INT功能.
- = 1, INT (外部中断输入脚), 在这种模式下, PORTB 的IOB0必须置“1”. IOB0作为I/O口输入功能通过硬件屏蔽了, 读取INT管脚信息的与读PORTB.方式相同

**WDTE :** WDT (watch-dog timer) 使能看门狗定时器

- = 0, 关闭WDT
- = 1, 使能WDT

## 2.1.9 WUCON (Port B输入改变/唤醒控制寄存器)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
09h (r/w)	WUCON	WUB7	WUB6	WUB5	WUB4	WUB3	WUB2	WUB1	WUB0

**WUB0** : =0, 禁止IOB0 输入改变/唤醒功能  
=1, 使能IOB0 输入改变/唤醒功能

**WUB1** : =0, 禁止IOB1 输入改变/唤醒功能  
=1, 使能IOB1 输入改变/唤醒功能

**WUB2** : =0, 禁止IOB2 输入改变/唤醒功能  
=1, 使能IOB2 输入改变/唤醒功能

**WUB3** : =0, 禁止IOB3 输入改变/唤醒功能  
=1, 使能IOB3 输入改变/唤醒功能

**WUB4** : =0, 禁止IOB4 输入改变/唤醒功能  
=1, 使能IOB4 输入改变/唤醒功能

**WUB5** : =0, 禁止IOB5 输入改变/唤醒功能  
=1, 使能IOB5 输入改变/唤醒功能

**WUB6** : =0, 禁止IOB6 输入改变/唤醒功能  
=1, 使能IOB6 输入改变/唤醒功能

**WUB7** : =0, 禁止IOB7 输入改变/唤醒功能  
=1, 使能IOB7 输入改变/唤醒功能

### 2.1.10 PCHBUF (PC指针高位缓冲区)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Ah (r/w)	PCHBUF	-	-	-	-	-	-	电脑2MSBs 缓冲	

**Bit1:Bit0** : 见2.1.3

**Bit7:Bit2** : 没有使用, 置 0

### 2.1.11 PDCON (I/O下拉控制寄存器)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Bh (r/w)	PDCON	*	/PDB2	/PDB1	/PDB0	/PDA3	/PDA2	/PDA1	/PDA0

**/PDA0** : = 0, 使能IOA0内部下拉  
= 1, 禁止IOA0内部下拉

**/PDA1** : = 0, 使能IOA1内部下拉  
= 1, 禁止IOA1内部下拉

**/PDA2** : = 0, 使能IOA2内部下拉  
= 1, 禁止IOA2内部下拉

**/PDA3** : = 0, 使能IOA3内部下拉  
= 1, 禁止IOA3内部下拉

**/PDB0** : = 0, 使能IOB 0内部下拉  
=1, 禁止IOB0内部下拉

**/PDB1** : = 0, 使能IOB1内部下拉  
=1, 禁止IOB1内部下拉

**/PDB2** : = 0, 使能IOB2内部下拉  
=1, 禁止IOB2内部下拉

**Bit7** : 没有使用, 置1

### 2.1.12 ODCON (I/O开漏控制寄存器)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Ch (r/w)	ODCON	ODB7	ODB6	ODB5	ODB4	-	ODB2	ODB1	ODB0

**ODB0** : = 0, 禁止IOB0内部开漏  
= 1, 使能 IOB0内部开漏

**ODB1** : = 0, 禁止IOB1内部开漏  
= 1, 使能 IOB1内部开漏

**ODB2** : = 0, 禁止IOB2内部开漏  
= 1, 使能 IOB2内部开漏

**Bit3** : 没有使用, 置0

**ODB4** : = 0, 禁止IOB4内部开漏  
= 1, 使能 IOB4内部开漏

**ODB5** : = 0, 禁止IOB5内部开漏  
= 1, 使能 IOB5内部开漏

**ODB6** : = 0, 禁止IOB6内部开漏  
= 1, 使能 IOB6内部开漏

**ODB7** : = 0, 禁止IOB7内部开漏  
= 1, 使能 IOB7内部开漏

### 2.1.13 PHCON (I/O上拉控制寄存器)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Dh (r/w)	PHCON	/PHB7	/PHB6	/PHB5	/PHB4	/PHB3	/PHB2	/PHB1	/PHB0

**/PHB0** : = 0, 使能IOB0内部上拉  
= 1, 禁止IOB0内部上拉

**/PHB1** : = 0, 使能IOB1内部上拉  
= 1, 禁止IOB1内部上拉

**/PHB2** : = 0, 使能IOB2内部上拉  
= 1, 禁止IOB2内部上拉

**/PHB3** : = 0, 使能IOB3内部上拉  
= 1, 禁止IOB3内部上拉

**/PHB4** : = 0, 使能IOB4内部上拉  
= 1, 禁止IOB4内部上拉

**/PHB5** : = 0, 使能IOB5内部上拉  
= 1, 禁止IOB5内部上拉

**/PHB6** : = 0, 使能IOB6内部上拉  
= 1, 禁止IOB6内部上拉

**/PHB7** : = 0, 使能IOB7内部上拉  
= 1, 禁止IOB7内部上拉

## 2.1.14 INTEN (中断屏蔽寄存器)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Eh (r/w)	INTEN	GIE	*	TC2IE	TC1IE	TC0IE	INTIE	PBIE	TOIE

**TOIE** : Timer0溢出中断屏蔽位。

- = 0, 禁止Timer0溢出中断
- = 1, 使能Timer0溢出中断

**PBIE** : Port B输入改变中断屏蔽位

- = 0, 禁止Port B输入改变中
- = 1, 使能Port B输入改变中

**INTIE** : 外部中断屏蔽位

- = 0, 禁止外部中断.
- = 1, 使能外部中断

**TC0IE** : TC0溢出中断屏蔽位。

- = 0, 禁止TC0溢出中断
- = 1, 使能TC0溢出中断

**TC1IE** : TC1溢出中断屏蔽位。

- = 0, 禁止TC1溢出中断
- = 1, 使能TC1溢出中断

**TC2IE** : TC2溢出中断屏蔽位。

- = 0, 禁止TC2溢出中断
- = 1, 使能TC2溢出中断

**Bit6** :没有使用, 置1

**GIE** : 中断允许总控位

- = 0, 禁止所有中断. 对于睡眠唤醒模式的中断事件, MCU将执行SLEEP后的指令。
- = 1, 使能所有没有屏蔽的中断. 对于睡眠唤醒模式的中断事件, MCU将跳转到中断地址 (008h)。

**注释:** 在中断事件发生时, GIEB被硬件清零并禁止一切中断, 所以GIE以及与该中断相关的中断屏蔽位需要重新开启。RETFIE 为退出中断程序并重新设置GIE =1允许中断。

## 2.1.15 INTFLAG (中断标志寄存器)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Fh (r/w)	INTFLAG	-	-	TC2IF	TC1IF	TC0IF	INTIF	PBIF	TOIF

**TOIF** : 溢出中断标志, 发生Timer0溢出中断置1, 软件设置清零

**PBIF** : Port B输入改变中断标志 interrupt flag. Port B输入改变时置1, 软件设置清零

**INTIF** : 外部中断标志. 当管脚INT上升沿/下降沿 (是上升沿/下降沿由 INTEDG 位 (OPTION<6>)决定) 时置1, 软件设置清零

**TC0IF** : 溢出中断标志, 发生TC0溢出中断置1, 软件设置清零

**TC1IF** : 溢出中断标志, 发生TC1溢出中断置1, 软件设置清零

**TC2IF** : 溢出中断标志, 发生TC2溢出中断置1, 软件设置清零

**Bit7:BIT6** : 没有使用, 置0

### 2.1.16 E2PCON( E2P控制寄存器 )

地址Bank1	名称	B7	B6	B5	B4	B3	B2	B1	B0
07h (r/w)	E2PCON	-	-	-	-	E2PCER	E2PSER	E2PBER	E2PWR

**E2PWR** : 内置EEPROM写使能, 置1开始写EEPROM, 烧写完自动清零

**E2PBER** : 内置EEPROM字节擦除使能, 置1开始按字节擦除EEPROM, 擦除完自动清零

**E2PSER** : 内置EEPROM扇区擦除使能, 置1开始按扇区擦除EEPROM, 擦除完自动清零

**E2PCER** : 内置EEPROM片擦除使能, 置1开始按片擦除EEPROM, 擦除完自动清零

**Bit7:BIT4** : 没有使用, 置0

### 2.1.17 E2PADR( E2P地址寄存器 )

地址Bank1	名称	B7	B6	B5	B4	B3	B2	B1	B0
08h (r/w)	E2PADR	E2PADR7	E2PADR6	E2PADR5	E2PADR4	E2PADR3	E2PADR2	E2PADR1	E2PADR0

**Bit7:BIT0**: 内置EEPROM地址, 地址00H~FFH共256字节

### 2.1.18 E2PDIN( E2P数据输入寄存器 )

地址Bank1	名称	B7	B6	B5	B4	B3	B2	B1	B0
09h (r/w)	E2PDIN	E2PDIN7	E2PDIN6	E2PDIN5	E2PDIN4	E2PDIN3	E2PDIN2	E2PDIN1	E2PDIN0

**Bit7:BIT0**: 内置EEPROM数据输入寄存器, 在烧写内置EEPROM前将要写的数据写入E2PDIN寄存器中

### 2.1.19 E2PDATA( E2P数据寄存器 )

地址Bank1	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Ah (r)	E2PDATA	E2PDA7	E2PDA6	E2PDA5	E2PDA4	E2PDA3	E2PDA2	E2PDA1	E2PDA0

**Bit7:BIT0**: 内置EEPROM数据读出寄存器, E2PADR所指地址中的数据, 只读寄存器, 用MOVR指令直接读入ACC中



### 2.1.20 TC0M( TC0控制寄存器 )

地址Bank2	名称	B7	B6	B5	B4	B3	B2	B1	B0
07h (r/w)	TC0M	TC0EN	TC0PS2	TC0PS1	TC0PS0	TC0CKS	ALOAD0	TC0OUT	PWM0EN

#### PWM0EN: PWM0输出控制

- = 0, 禁止PWM0输出, IOB0为GPIO引脚
- = 1, 使能PWM0输出, IOB0输出PWM0信号, 其占空比由TC0OUT和ALOAD0控制

#### TC0OUT: TC0溢出输出信号控制, 仅当PWM0EN=0时有效

- = 0, 禁止, IOB0为GPIO引脚
- = 1, 使能, IOB0输出TC0OUT信号

#### ALOAD0: 自动装载控制, 仅当PWM0EN=0时有效

- = 0, 禁止TC0自动装载
- = 1, 使能TC0自动装载

#### TC0CKS: TC0时钟选择

- = 0, TC0 选择Fcpu作为时钟输入
- = 1, TC0 选择Fhosc作为时钟输入

#### TC0PS[2:0]: TC0分频选择位

- = 000, Ftc0/128
- = 001, Ftc0/64
- = 010, Ftc0/32
- = 011, Ftc0/16
- = 100, Ftc0/8
- = 101, Ftc0/4
- = 110, Ftc0/2
- = 111, Ftc0/1

#### TC0EN: TC0启动控制位

- = 0, 禁止TC0定时器
- = 1, 开启TC0定时器

### 2.1.21 TC0C( TC0计数寄存器 )

地址Bank2	名称	B7	B6	B5	B4	B3	B2	B1	B0
08h (r/w)	TC0C	TC0C7	TC0C6	TC0C5	TC0C4	TC0C3	TC0C2	TC0C1	TC0C0

8位计数器TC0C溢出时, TC0IF置1并由程序清零, 用来控制TC0的中断间隔时间。首先须写入正确的值到TC0C和TC0R寄存器, 并使能TC0定时器以保证第一个周期正确。TC0溢出后, TC0R的值自动装入TC0C。

#### TC0C初始值的计算公式如下:

$$TC0C初始值 = N - (TC0中断间隔时间 * TC0时钟rate)$$

N为TCX二进制计数范围。各模式下参数的设定如下表所示:

PWM0EN	ALOAD0	TC0OUT	N	TC0C范围	注释
0	x	x	256	00H~FFH	每计数256次溢出
1	0	0	256	00H~FFH	每计数256次溢出
1	0	1	64	00H~3FH	每计数64次溢出
1	1	0	32	00H~1FH	每计数32次溢出
1	1	1	16	00H~0FH	每计数16次溢出

### 2.1.22 TC0R( TC0自动装载寄存器 )

地址Bank2	名称	B7	B6	B5	B4	B3	B2	B1	B0
09h (r/w)	TC0R	TC0R7	TC0R6	TC0R5	TC0R4	TC0R3	TC0R2	TC0R1	TC0R0

TC0内置自动重装功能，TC0R寄存器存储重装值。TC0C溢出时，TC0R的值自动装入TC0C中。TC0定时器工作在计时模式时，要通过修改TC0R寄存器来修改TC0的间隔时间，而不是通过修改TC0C寄存器。在TC0定时器溢出后，新的TC0C值会被更新，TC0R会将新的值装载到TC0C寄存器中。但在初次设置TC0M时，必须要在开启TC0定时器前把TC0C以及TC0R设置成相同的值。

TC0为双重缓存器结构。若程序对TC0R进行了修改，那么修改后的TC0R值首先被暂存在TC0R的第一个缓存器中，TC0溢出后，TC0R的新值就会被存入TC0R缓存器中，从而避免TC0中断时间出错以及PWM0误动作。

### 2.1.23 TC1M( TC1控制寄存器 )

地址Bank2	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Ah (r/w)	TC1M	TC1EN	TC1PS2	TC1PS1	TC1PS0	TC1CKS	ALOAD1	TC1OUT	PWM1EN

具体操作请参考TC0M

### 2.1.24 TC1C( TC1计数寄存器 )

地址Bank2	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Bh (r/w)	TC1C	TC1C7	TC1C6	TC1C5	TC1C4	TC1C3	TC1C2	TC1C1	TC1C0

具体操作请参考TC0C

### 2.1.25 TC1R( TC1自动装载寄存器 )

地址Bank2	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Ch (r/w)	TC1R	TC1R7	TC1R6	TC1R5	TC1R4	TC1R3	TC1R2	TC1R1	TC1R0

具体操作请参考TC0R

### 2.1.26 TC2M( TC2控制寄存器 )

地址Bank2	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Dh (r/w)	TC2M	TC2EN	TC2PS2	TC2PS1	TC2PS0	TC2CKS	ALOAD2	TC2OUT	PWM2EN

具体操作请参考TC0M

### 2.1.27 TC2C( TC2计数寄存器 )

地址Bank2	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Eh (r/w)	TC2C	TC2C7	TC2C6	TC2C5	TC2C4	TC2C3	TC2C2	TC2C1	TC2C0

具体操作请参考TC0C

### 2.1.28 TC2R( TC2自动装载寄存器 )

地址Bank2	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Fh (r/w)	TC2R	TC2R7	TC2R6	TC2R5	TC2R4	TC2R3	TC2R2	TC2R1	TC2R0

具体操作请参考TC0R

### 2.1.29 ACC (Accumulator)累加器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
N/A (r/w)	ACC	累加器							

累加器是一个内部数据转化、指令操作和存放操作结果的存储单元，不能被访问。

### 2.1.30 OPTION Register (选项寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
N/A (w)	OPTION	T0TB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

通过OPTION 指令访问

在执行OPTION 指令时候，该数据单元由ACC（累加器）转化为选项寄存器（OPTION Register）。

选项寄存器是一个7位只写寄存器，它的一些控制位主要用来配置与Timer0/WDT 分频器，Timer0，外部中断选项相关信息。除INTEDG位以外其他位是只写并可以置1。

#### PS2:PS0 : 分频率选择控制位

PS2:PS0	Timer0 Rate	WDT Rate
0 0 0	1:2	1:1
0 0 1	1:4	1:2
0 1 0	1:8	1:4
0 1 1	1:16	1:8
1 0 0	1:32	1:16
1 0 1	1:64	1:32
1 1 0	1:128	1:64
1 1 1	1:256	1:128

#### PSA : 分频器选择位.

- = 1, WDT (看门狗定时器)
- = 0, TMR0 (Timer0)

#### T0SE : TMR0触发方式控制位

- = 1, T0CKI脚下降沿触发计数
- = 0, T0CKI脚上升沿触发计数

#### T0CS : TMR0 时钟源选择控制位

- = 1, 外部T0CKI脚. 当IOST IOB2 = "0".时, IOB2/T0CKI脚设置为输入
- = 0, 内部指令时钟周期或者RTC时钟

#### INTEDG :中断触发方式控制位.

- = 1, 中断触发方式为INT脚上升沿出发
- = 0, 中断触发方式为INT脚下下降沿出发

#### T0TB :T0时钟RTC选择位.

- = 1, T0时钟选择RTC时钟（外部32768晶振）
- = 0, T0时钟选择内部指令时钟

### 2.1.31 IOSTA & IOSTB (I/O口控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
N/A (w)	IOSTA					IOSTA3	IOSTA2	IOSTA1	IOSTA0
N/A (w)	IOSTB	IOSTB7	IOSTB6	IOSTB5	IOSTB4	IOSTB3	IOSTB2	IOSTB1	IOSTB0

通过IOST指令访问

通过指令IOST R (05h~06h)把累加器A的内容加载到I/O控制寄存器，按位将IOSTA，IOSTB设为1表示该脚为输入（高阻抗）、设为0时表示该脚为输出。

IOST寄存器只写，系统复位以后设置为输入（高阻抗）。

## 2.2 I/O Ports

Port A 和 PortB为双向三态I/O 口。Port A为4脚I/O口。Port B 为8脚I/O口。IOB2需要通过选项寄存器(Option)的T0CS(OPTION<5>)位控制,所有的I/O的输入/输出方式由I/O控制寄存器(IOSTA, IOSTB)设置。IOB<7:4>和IOB<2:0>有相应的上拉控制位(PHCON 寄存器)来设置使能内部上拉,如果设置为输出模式,内部上拉功能会自动关闭。IOA<3:0>和IOB<2:0>有相应的下拉控制位(PDCON寄存器)来设置使能内部下拉,如果设置为输出模式,内部下拉功能会自动关闭。IOB<7:4>和IOB<2:0>有相应的开漏控制位(ODCON寄存器)来设置使能开漏来设置输出为开漏输出。

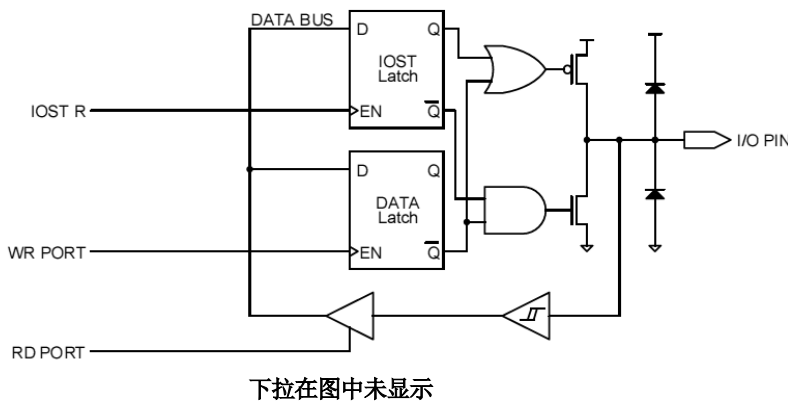
IOB<7:0> 有输入改变中断/唤醒功能.它的每个管脚是否具有该功能通过取决于WUCON寄存器的相应位。

当EIS(PCON<6>)=1 时, IOB0作为外部中断输入脚,在该模式下IOB0 输入改变中断/唤醒功能被硬件屏蔽,即使软件已经设置为中断/唤醒功能可用也不可启用该功能。

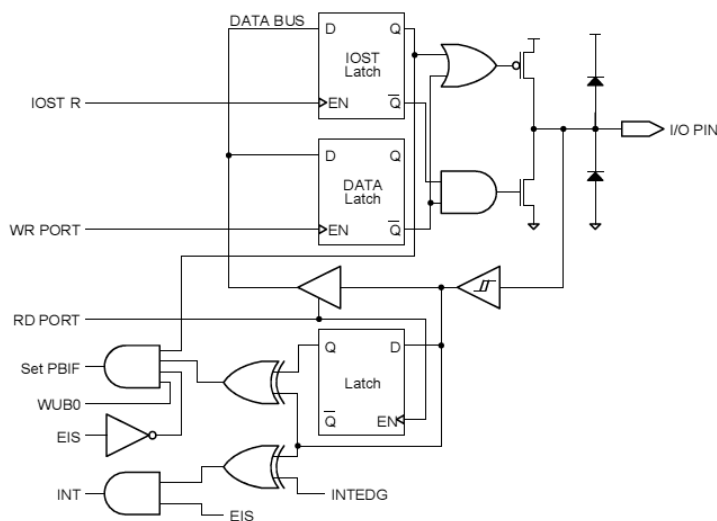
配置字能交替设置I/O口的不同功能,功能交替设置完以后,读的I/O的值为0。

图2.3: I/O 脚的结构图

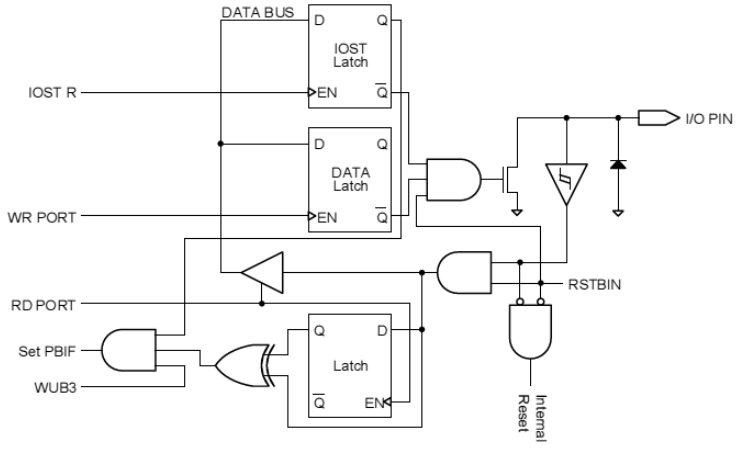
IOA3 ~ IOA0:



IOB0/INT:

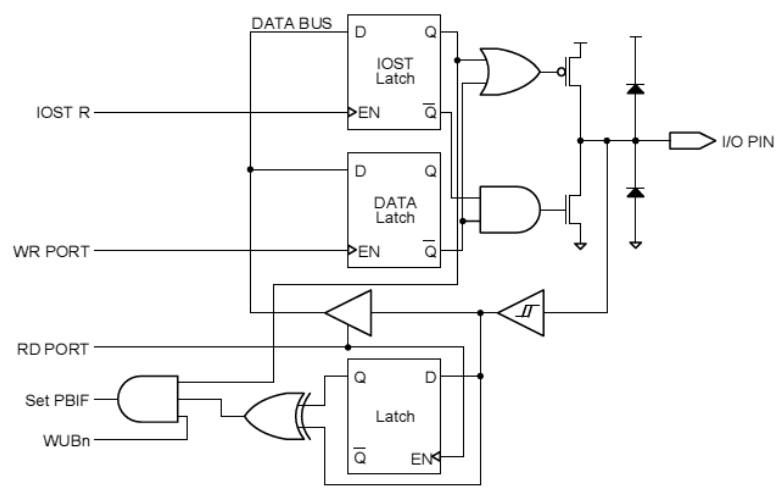


IOB3:



电压在这个引脚禁止超过VDD

IOB7 ~ IOB4, IOB2 ~ IOB1:



上拉/下拉和漏极开漏在图中未显示

## 2.3 Timer0/WDT & Prescaler/TC0/TC1/TC2

### 2.3.1 Timer0

Timer0为8位定时/计数器， Timer0 的时钟源可以是内部或外部时钟源(T0CKI pin)

#### 2.3.1.1 使用内部时钟: 定时模式

T0CS(OPTION<5>)=0为定时模式， 定时模式在没有预置器的情况下， 定时寄存器每个指令周期自动加1， 设置TMR0以后， 定时器将在两个时钟周期以后开始自增。

#### 2.3.1.2使用外部时钟: 计数模式

T0CS(OPTION<5>)=1为计数模式， 是选择通过T0CK管脚的上升或下降沿触发Timer0寄存器的增加由T0SE 位(OPTION<4>)决定， 外在时钟要求与内部时钟(Tosc)同步。同步以后， Timer0实际增加有一个延迟。

在没有预置器的情况下， 外部时钟输入同样也可以作为预置器输出； T0CKI与内部时钟同步时能方便处理在T2 和 T4周期上的预分频。因此T0CKI为高或低电平必须要保持两个以上时钟周期才有效。

有预置分频器， 外部时钟输入被异步分频器平分， 这种常用来计算波形。因此:因此T0CKI的一个波形周期至少4Tosc才能被 预置器平分。

### 2.3.2 看门狗定时器 (WDT)

看门狗定时器 (WDT) 的运行依赖于芯片里的RC振荡器， 无需任何额外电路即能工作。不管时钟OSCI和OSCO管脚是否关闭， 它都能运行， 如在睡眠模式。在一般操作或睡眠模式情况下， 看门狗定时器的溢出都会导致MCU复位同时TO (STATUS<4>)位被清零。如WDTE 位(PCON<7>)清零。看门狗定时器不能工作。

在没有预置器时看门狗的溢出为18 ms， 4.5ms， 288ms， 72ms这个时间可以通过SUT<1:0> 设置。

需要看门狗的溢出周期变长可以通过设置OPTION寄存器的看门狗定时器分频大于1:128.， 因此最长的看门狗溢出周期为 36.8 秒。.

CLRWDT指令能使WDT和预置器清零， 启用看门狗可以防止超时， 如果超时MCU能复位。

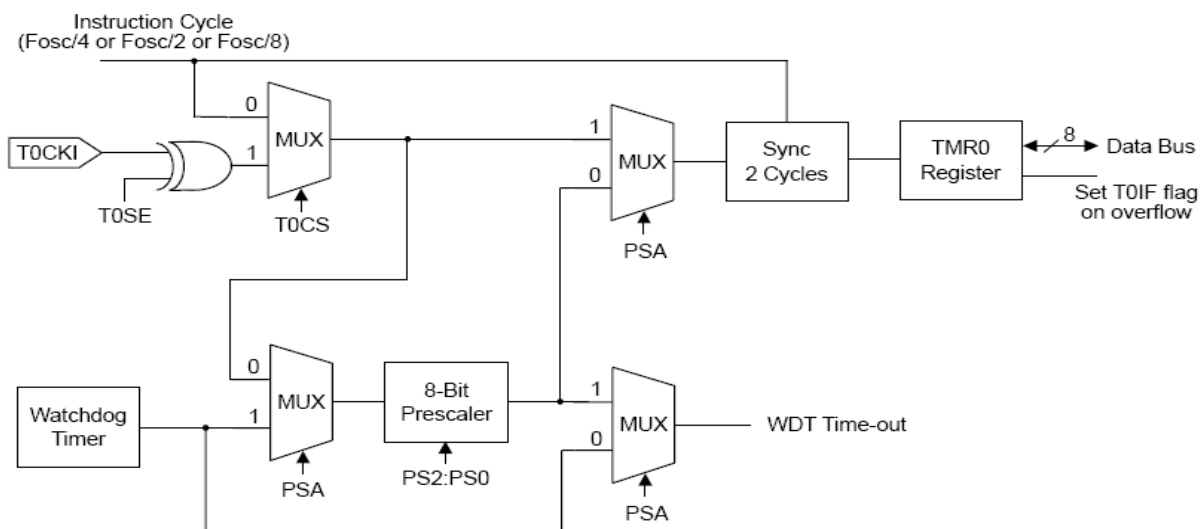
SLEEP 指令重置WDT和预置器， 启用看门狗就给机器分派了一个最大睡眠时间。

### 2.3.3 Prescaler (预置器)

有一个8位的向下计数器作为Timer0和看门狗定时器(WDT)的预置器。注意该预置器只能分配给Timer0 或 WDT使用， 不能两者同时使用。PSA 位(OPTION<3>) 决定预置器是指派给Timer0还是WDT。PS<2:0> 位(OPTION<2:0>) 配置分频。当作为Timer0的预置器的时候， TMR0会被预置器清零。当作为WDT的预置器的时候， CLRWD 指令会清除预置器内容。预置器不能读写， 机器复位， 预置器各位全为1。

为了避免机器非正常复位， 当Timer0 或 WDT的预置器发生改变的时候， 需要执行CLRWDT 或 CLRR TMR0 指令， 反之亦然。

图2.4: Timer0/WDT Prescaler结构图



### 2.3.4 TC0

8位二进制定时/计数器具有基本定时器、Buzzer和PWM功能。基本定时器功能可以支持标志显示（TC0IF）和中断操作（中断向量）。由TC0M、TC0C、TC0R寄存器控制TC0的中断间隔时间。TC0还内置Buzzer和PWM功能，Buzzer和PWM的周期和分辨率由TC0PS[2:0]、TC0R寄存器控制，故具有良好性能的Buzzer和PWM可以处理IR载波信号，马达控制和光度调节等。

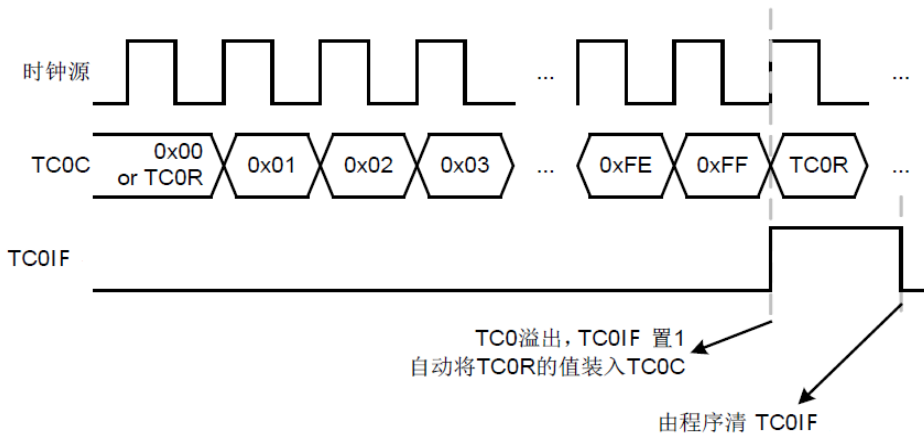
TC0的主要用途如下：

1. 8位可编程定时器：根据选择的时钟信号，产生周期中断；
2. 中断功能：TC0定时器支持中断，当TC0溢出时，TC0F置1，当TC0IE=1和GIE=1时系统执行中断；
3. PWM输出：由TC0rate和TC0R寄存器控制占空比/周期；
4. Buzzer输出：Buzzer的输出信号是TC0间隔时间的1/2个周期；

#### 2.3.4.1 TC0定时模式

TC0定时器由TC0EN控制。当TC0EN=0时，TC0停止工作；当TC0EN=1时，TC0开始计数。使能TC0之前，先要设定好TC0的功能模式，如基本定时器、TC0中断等。TC0C溢出（从0FFH到00H）时，TC0IF置1以显示溢出状态并由程序清零。在不同的功能模式下，TC0C不同的值对应不同的操作，若改变TC0C的值影响到操作，会导致功能出错。TC0内置双重缓存器以避免此种状况的发生。在TC0C计数的过程中不断的刷新TC0C，保证将最新的值存入TC0R（重装缓存器）中，当TC0溢出后，TC0R的值由自动存入TC0C。进入下一个周期后，TC0进入新的工作状态。定时器模式下，由ALOAD0控制自动重装功能；PWM模式下，使能TC0时，自动使能TC0的自动重装功能。如果使能TC0中断功能（TC0IE=1），在TC0溢出时系统执行中断服务程序，在中断时必须由程序清TC0IF。

图2.5:TC0定时溢出示意图

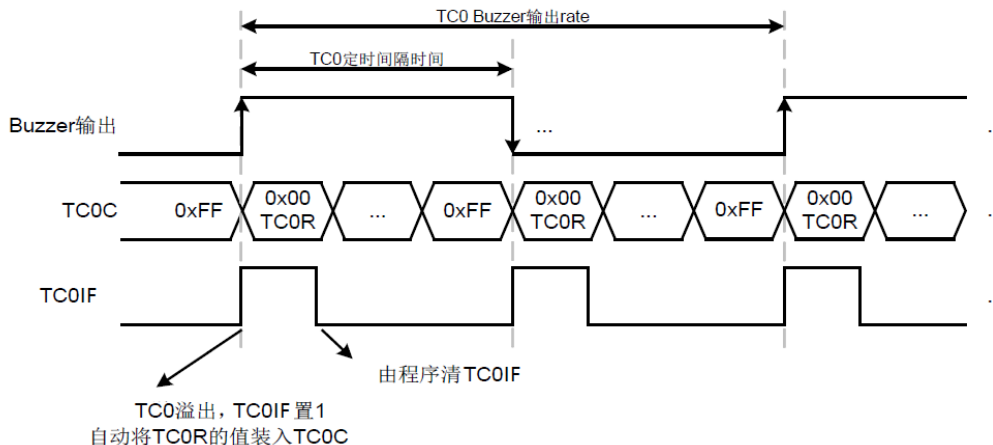


TC0PS[2:0]	TC0时钟	TC0间隔时间			
		Fhosc=16MHz Fcpu=Fhosc/4 (4T)		Fhosc=4Mhz Fcpu=Fhosc/4 (4T)	
		max.(ms)	Unit(us)	max.(ms)	Unit(us)
000b	Ftc0/128	8.192	32	32.768	128
001b	Ftc0/64	4.096	16	16.384	64
010b	Ftc0/32	2.048	8	8.192	32
011b	Ftc0/16	1.024	4	4.096	16
100b	Ftc0/8	0.512	2	2.048	8
101b	Ftc0/4	0.256	1	1.024	4
110b	Ftc0/2	0.128	0.5	0.512	2
111b	Ftc0/1	0.064	0.25	0.256	1

### 2.3.4.2 TC0 BUZZER输出

Buzzer输出是一个简单的1/2占空比信号输出，由TC0产生。当TC0溢出时，Buzzer开始输出一个方波，中断间隔时间频率2分频后作为Buzzer输出的频率。Buzzer输出的波形图如下所示：

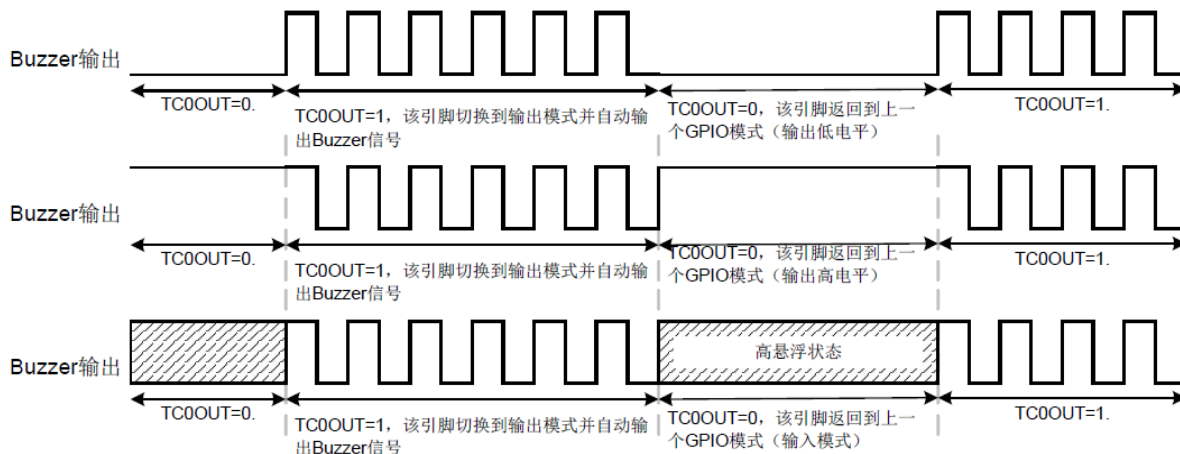
图2.6: Buzzer输出示意图



TC0溢出后，Buzzer输出时，TC0IF有效，且当TC0IE=1时，使能TC0中断功能。但强烈建议小心同时使用Buzzer和TC0定时器，以确保两种功能都能正常工作。

Buzzer输出引脚与GPIO引脚共用，TC0OUT=1时，该引脚自动设为Buzzer输出引脚。如清TC0OUT位以禁止Buzzer输出后，该引脚自动返回到最后一个GPIO模式。

图2.7: Buzzer输出/IO切换状态示意图



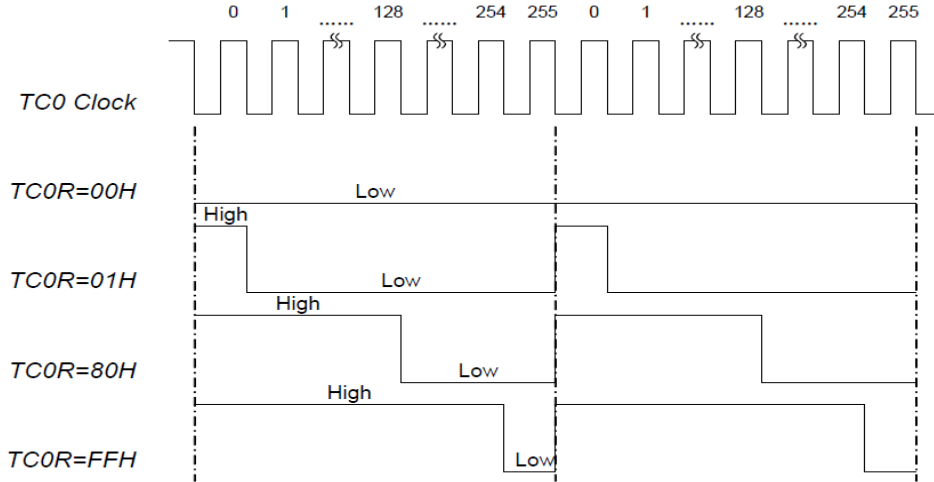


### 2.3.4.2 TC0 PWM输出（脉宽调制）

可编程控制占空比/周期的PWM可以提供不同的PWM信号。使能TC0定时器且PWM0EN=1时，由PWM0输出引脚（IOB0）输出PWM信号。PWM首先输出高电平，然后输出低电平。TC0PS[2:0]控制PWM的周期，ALOAD0和TC0OUT决定PWM的分辨率，TC0R寄存器决定PWM的占空比（脉冲高电平的长度）。开启TC0定时器且定时器溢出后，TC0C的初始值为0。当TC0C=TC0R时，PWM输出低电平；TC0溢出时（TC0C的值从0FFH到00H），整个PWM周期完成，并进入下一个周期。TC0溢出时，PWM的一个周期完成。在PWM输出的过程由程序更改PWM的占空比，则在下一个周期开始输出新的占空比的PWM信号。

图2.8:PWM输出示意图

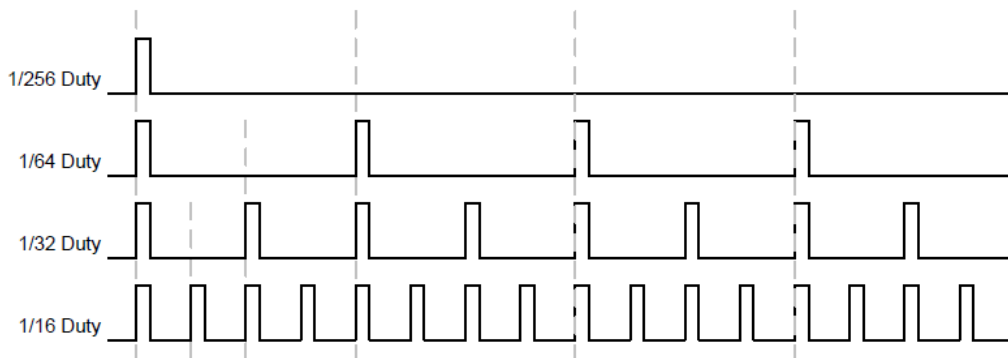
PWM 输出占空比随 TC0R 的变化而变化：0/256~255/256。



PWM的分辨率由ALOAD0和TC0OUT决定，以实现高速PWM信号。当ALOAD0、TC0OUT = 00时，PWM的分辨率为1/256；ALOAD0、TC0OUT = 01时，PWM的分辨率为1/64；ALOAD0、TC0OUT = 10时，PWM的分辨率为1/32；ALOAD0、TC0OUT = 11时，PWM的分辨率为1/16。若需调制PWM的分辨率，TC0R PWM的占空比控制范围必须调制到一个合适的分辨率。PWM输出过程中，TC0溢出时，TC0IF有效，TC0IE=1时，则使能TC0中断。但强烈建议小心同时使用PWM和TC0定时器功能，保证两种功能都能正常工作。

图2.9:PWM周期设置示意图

ALOAD0	TC0OUT	PWM 分辨率	TC0R 有效值	TC0R 有效值（二进制）
0	0	256	00H~FFH	00000000b~11111111b
0	1	64	00H~3FH	xx000000b~xx111111b
1	0	32	00H~1FH	xxx00000b~xxx11111b
1	1	16	00H~0FH	xxxx0000b~xxxx1111b



PWM输出引脚和GPIO引脚共用，PWM0OUT=1时，该引脚自动输出PWM信号。如果清PWM0OUT位以禁止PWM时，该引脚返回到最后一个GPIO模式。

### 2.3.5 TC1/TC2

所有操作具体参考TC0

## 2.4 中断方式

SM701系统具备有六种中断方式:

1. INT 管脚的外部中断
2. TMR0 溢出中断
3. Port B 输入改变中断 (IOB7:IOB0脚)
4. TC0 溢出中断
5. TC1 溢出中断
6. TC2 溢出中断

INTFLAG为中断标志寄存器, 决定该寄存器机器所发生的中断状态。

中断允许总控位GIE (INTEN<7>), 能使所有中断被开放 (GIE=1) 或屏蔽所有中断 (GIE=0), 每中断能否启用决定INTEN 寄存器同时保证GIE=1。

中断发生时GIE 位 (在中断发生前GIE 位和该中断相关的中断屏蔽位置1) 被硬件清零从而禁止进一步中断 (SM701不区分中断优先级), 同时下条指令跳到008h后开始执行。中断标志位在中断允许总控位GIE重新置1的时候需要被软件清零以防止重复中断。一个中断标志位 (PBIF除外的) 会被它的中断事件置1, 而不管与它相关的中断屏蔽位是否启用。通过INTFLAG 和INTEN 的相应中位来判断是否发生中断以及中断类型。当通过INT指令发生软中断时, 下条指令跳到002后开始执行。

### 2.4.1 外部中断

外部中断INT管脚上升沿还是下降沿触发由 INTEDG 位 (OPTION<6>)决定, 当一个有效的跳变发生时标志位INTIF置1, 如INTIE位 (INTEN<2>)清零, 该中断被屏蔽。

在睡眠之前INTIE 位已被置1, INT管脚可以作为系统睡眠条件。在睡眠之前GIE位已被置1机器唤醒以后会执行中断服务程序, 否则会运行睡眠以后的下一条指令。

### 2.4.2 Timer0 中断

TMR0发生溢出 (FFh → 00h)时T0IF标志位置1 (INTFLAG<0>). T0IE 位(INTEN<0>)清零, 该中断被屏蔽。

### 2.4.3 Port B 输入改变中断

输入改变中断触发时IOB<7:0> PBIF标志位置1 (INTFLAG<1>). PBIE位(INTEN<1>)清零, 该中断被屏蔽。

在输入改变中断发生之前, 必须读取port B信息

与PortB的管脚相对应的WUBn位 (WUCON<7:0>) i;清零或设置为输出或IOB0 脚设置为外部中断输入脚INT 都拥有该功能。PBIE在睡眠之前置1, Port B 输入脚改变中断也可以作为睡眠唤醒条件。在睡眠之前GIE位已被置1机器唤醒以后会执行中断服务程序, 否则会运行睡眠以后的下一条指令。

### 2.4.4 TC0 中断

TC0发生溢出 (FFh → 00h)时TC0IF标志位置1 (INTFLAG<3>). TC0IE 位(INTEN<3>)清零, 该中断被屏蔽。

### 2.4.5 TC1 中断

TC1发生溢出 (FFh → 00h)时TC1IF标志位置1 (INTFLAG<4>). TC1IE 位(INTEN<4>)清零, 该中断被屏蔽。

### 2.4.6 TC2 中断

TC2发生溢出 (FFh → 00h)时TC2IF标志位置1 (INTFLAG<5>). TC2IE 位(INTEN<5>)清零, 该中断被屏蔽。

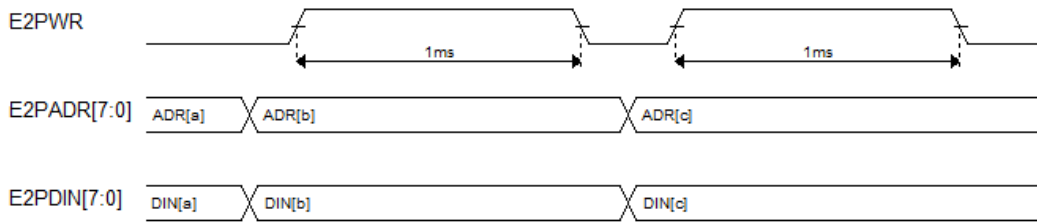
## 2.5 内置EEPROM操作

内置EEPROM共有256个字节，分为16个扇区，每个扇区以及每个字节都可以单独擦除，也可以整个地址全部擦除。E2PADR寄存器为内置EEPROM的地址，通过给E2PADR写入不同的值来选择不同的地址。E2PDIN寄存器为即将要写入内置EEPROM的值，通过给E2PDIN写入不同的值来修改将要写入的数据。内置EEPROM只能按字节写入数据，写入时间大约为1ms。擦除时间大约为100ms。

### 2.5.1 内置EEPROM烧写

将需要烧写的地址写入E2PADR寄存器，将需要烧写的的数据写入E2PDIN寄存器，E2PCON控制寄存器中的E2PWR位置1，内置EEPROM开始烧写，大约1ms左右烧写结束，烧写完毕后E2PWR位自动清零。

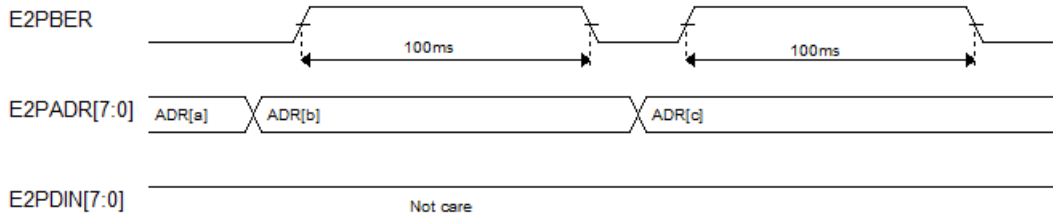
图2.10:内置EEPROM烧写示意图



### 2.5.2 内置EEPROM的字节擦除

将需要擦除的地址写入E2PADR寄存器，E2PCON控制寄存器中的E2PBER位置1，内置EEPROM开始擦除，大约100ms左右擦除结束，擦除完毕后E2PBER位自动清零。

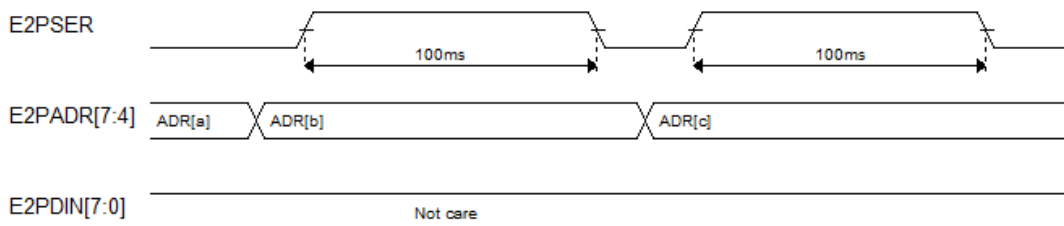
图2.11:内置EEPROM字节擦除示意图



### 2.5.3 内置EEPROM的扇区擦除

将需要擦除的地址写入E2PADR寄存器，扇区地址由E2PADR高4位决定，E2PCON控制寄存器中的E2PSER位置1，内置EEPROM开始擦除，大约100ms左右擦除结束，擦除完毕后E2PSER位自动清零。

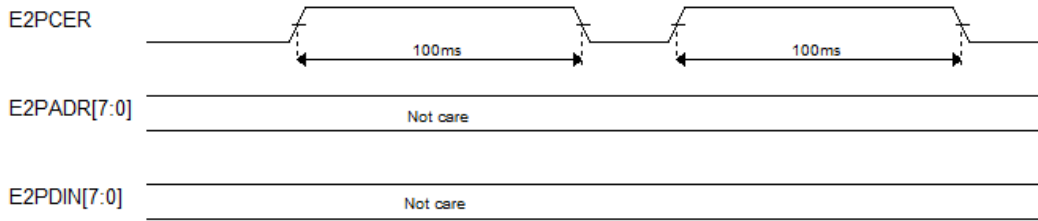
图2.12:内置EEPROM扇区擦除示意图



### 2.5.4 内置EEPROM的片擦除

与地址无关，所有地址中的数据将被擦除，E2PCON控制寄存器中的E2PCER位置1，内置EEPROM开始擦除，大约100ms左右擦除结束，擦除完毕后E2PCER位自动清零。

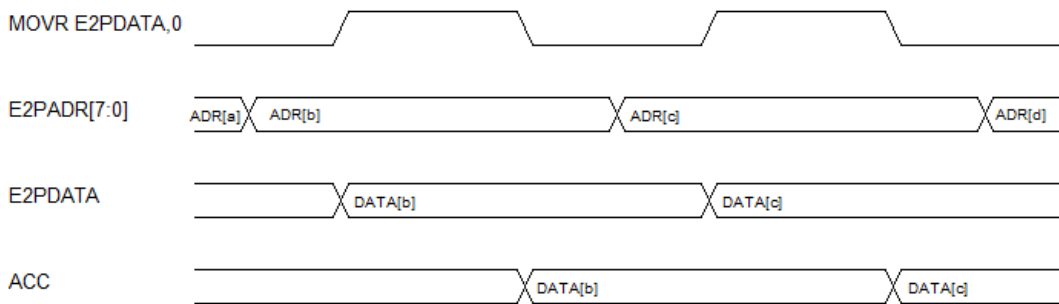
图2.13:内置EEPROM片擦除示意图



### 2.5.5 内置EEPROM的读操作

将需要读出数据的地址写入E2PADR寄存器，然后通过MOVR指令读出E2PDATA寄存器中的数据，数据存入ACC中。E2PDATA是只读寄存器。

图2.14:内置EEPROM读操作示意图



## 2.6 省电模式 (SLEEP)

执行SLEEP 指令以后机器进入省电模式。

执行SLEEP 指令， /PD 位清零 (STATUS<3>)， /TO位置1，看门狗清零同时保持运行状态，晶体停振。

I/O维持原状

### 2.6.1 睡眠唤醒

在睡眠状态下，单片机能通过以下方式唤醒：

1. RSTB 管脚复位
2. 看门狗复位 (机器设置了看门狗)。
3. RB0/INT管脚中断，或PORTB 输入改变中断。

外部的RSTB管脚和看门狗溢出都能使机器复位。通过查看 /PD 和/TO 位可以检测机器是哪种复位，/PD位置1为上电复位，置0为执行SLEEP， /TO 位置0为看门狗溢出复位。

机器通过中断唤醒，该中断屏蔽位置1，中断唤醒不管GIE是否置1。当GIE位被清零，机器唤醒以后执行SLEEP指令以后的指令；当GIE位被置1，机器唤醒以后跳转到中断复位地址 (008h)。在高频或低频模式机器复位延迟时间为18/4.5/288/72ms (该延迟时间由SUT<1:0>设置) 加上64个振荡周期。

在IRC/ERIC or ERC 模式，机器复位延迟时间为640us。

## 2.7 复位

SM701单片机能通过以下方式复位：

1. 上电复位(POR)
2. 掉电复位(Brown-out Reset BOR)
3. RSTB 管脚复位
4. 看门狗WDT溢出复位

一些寄存器在一些复位条件下没有影响，在上电和其他一些复位情况下它们的状态是未知的。。大多数寄存器会回到复位状态在上电复位，RSTB 管脚复位，看门狗WDT溢出复位。

对Vdd上升信号检测告之芯片是否加上上电复位脉冲信号。要使用这个特点，用户需要把RSTB管脚连接到Vdd。

掉电复位作为一种典型应用主要用在 AC 或重载交换的应用上。

芯片上的低电压检测模块到电压低于一个固定的电压也会使芯片复位，这样能保证芯片只能在正常电压范围内工作。

RSTB或WDT睡眠唤醒也导致芯片复位，其复位操作的不会在睡眠之前。

根据不同的复原状态设置对/TO和/PD位(STATUS<4 :3>)置1或清零。

### 2.7.1 上电复位计数器(Power-up Reset Timer PWRT)

上电复位计数器提供一个 18/4.5/288/72ms 延迟时间 (该延迟时间由 SUT<1:0>设置) (或 640us，基于不同的振荡源和复位条件) 在 Power-on Reset (POR)， Brown-out Reset (BOR)， RSTB Reset 或 看门狗溢出复位。只要 PWRT 在运行，设备就一直保持的复位状态。Vdd、温度和其他变化而会影响 PWDT 控制的设备延迟时间。

表2.1: PWRT Period

Oscillator Mode	Power-on Reset Brown-out Reset	RSTB Reset WDT time-out Reset
ERC & IRC/ERIC	18/4.5/288/72 ms	640 us
HF & LF & IRC_RTC	18/4.5/288/72 ms	18/4.5/288/72ms

### 2.7.2 振荡启动计数器(Oscillator Start-up Timer OST)

在HF或 LF或IRC\_RTC振荡模式下在PWRT 延迟 (18/4.5/288/72ms) 之后振荡启动计数器会再提供一个64个clock的延迟。这种延迟晶体谐振器能提供稳定的振荡源，这段时间内只要OST在工作，设备就一直保持的复位状态。

在 OSCI 信号的振幅到达振荡器输入最大振幅之后，该计数器只开始增加。

### 2.7.3 复位顺序

SM701复位时序如下：

1. 复位锁存器置1，PWRT & OST 清零。
2. 当内部的POR，BOR，RSTB 复位或 WDT溢出复位脉冲加载完成后，PWRT开始计数。
3. PWRT溢出以后，OST开始计数延迟。
4. OST延迟完成以后，复位锁存器清零最后芯片得到一个复位信号。

在高频或低频振荡模式机器复位延迟时间为18/4.5/288/72ms加上64个振荡周期，在IRC/ERIC，ERC振荡模式单机会在Power-on Reset (POR)，Brown-out Reset (BOR)，或RSTB复位以后在延迟640us，看门狗溢出复位后再延迟18/4.5/288/72ms的时间。

图2.15: 复位电路结构图

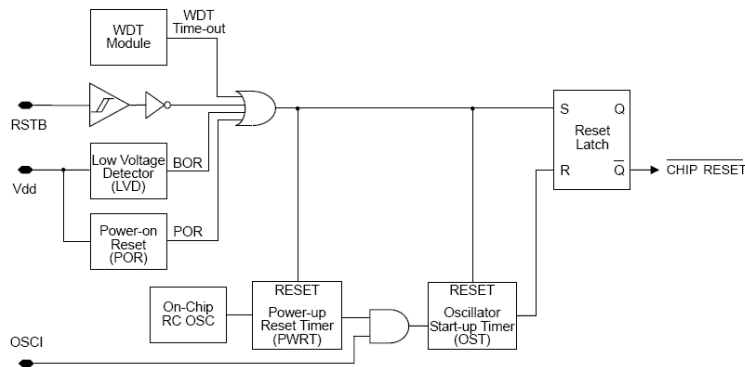


表2.2: 复位以后各个寄存器状态列表

寄存器	地址	上电复位 掉电复位	RSTB复位 WDT 复位
ACC	N/A	xxxx xxxx	uuuu uuuu
OPTION	N/A	0011 1111	0011 1111
IOSTA	N/A	---- 1111	---- 1111
IOSTB	N/A	1111 1111	1111 1111
INDF	00h	xxxx xxxx	uuuu uuuu
TMR0	01h	xxxx xxxx	uuuu uuuu
PCL	02h	1111 1111	1111 1111
STATUS	03h	0001 1xxx	000# #uuu
FSR	04h	00xx xxxx	00uu uuuu
PORTA	05h	---- xxxx	---- uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu
General Purpose Registers	10 ~ 3Fh	xxxx xxxx	uuuu uuuu

Bank0	地址	上电复位 掉电复位	RSTB复位 WDT 复位
TCXCON	07h	0000 0000	0000 0000
PCON	08h	1010 0000	1010 0000
WUCON	09h	0000 0000	0000 0000
PCHBUF	0Ah	---- --00	---- --00
PDCON	0Bh	1111 1111	1111 1111
ODCON	0Ch	0000 0000	0000 0000
PHCON	0Dh	1111 1111	1111 1111
INTEN	0Eh	0-00 0000	0-00 0000
INTFLAG	0Fh	--00 0000	--00 0000

Bank1	地址	上电复位 掉电复位	RSTB复位 WDT 复位
E2PCON	07h	---- 0000	---- 0000
E2PADR	08h	0000 0000	0000 0000
E2PDIN	09h	0000 0000	0000 0000
E2PDATA	0Ah	uuuu uuuu	uuuu uuuu

Bank2	地址	上电复位 掉电复位	RSTB复位 WDT 复位
TC0M	07h	0000 0000	0000 0000
TC0C	08h	0000 0000	0000 0000
TC0R	09h	0000 0000	0000 0000
TC1M	0Ah	0000 0000	0000 0000
TC1C	0Bh	0000 0000	0000 0000
TC1R	0Ch	0000 0000	0000 0000
TC2M	0Dh	0000 0000	0000 0000
TC2C	0Eh	0000 0000	0000 0000
TC2R	0Fh	0000 0000	0000 0000

Legend: u = 不变, x = 未知, - = 不起作用, # = 参见下表的值

表2.3: RST/TO / PD 复位和唤醒后的状态

RST	/TO	/PD	复位方式
0	1	1	Power-on Reset
0	1	1	Brown-out reset
0	u	u	RSTB Reset during normal operation
0	1	0	RSTB Reset during SLEEP
0	0	1	WDT Reset during normal operation
0	0	0	WDT Wake-up during SLEEP
1	1	0	Wake-up on pin change during SLEEP

Legend: u =不变

表2.4: /TO /PD状态位影响事件

事件	/TO	/PD
Power-on	1	1
WDT Time-Out	0	u
SLEEP instruction	1	0
CLRWDT instruction	1	1

Legend: u =不变

## 2.8 十六进制转化为十进制 (Hexadecimal Convert to Decimal HCD)

SM701具有十进制格式化功能. 当一个寄存器里面的内容需要十进制转化的时候, 在执行操作ALU以后必须把结果进行相应的进制转化。一个数据在处理过程中进行了转化成了十进制, 那么所有对这个数进行的操作(包含存放该数据的RAM单元, accumulator (ACC), 立即数, 以及所要查表信息)都的进行十进制转化, 这样的运算结果才正确。DAA指令能在加法运算完成以后将ACC 里的数据从十六进制转化为十进制重存给ACC 转换操作在例子 2.2 中被说明。

例2.2: DAA 转化

Address	Code
NA	#include <FC701.ASH>
n	...
n+1	MOVIA 0x90 ;Set immediate data = decimal format number "90" (ACC ← 90h)
n+2	MOVAR 0x30 ;Load immediate data "90" to data memory address 30H
n+3	MOVIA 0x10 ;Set immediate data = decimal format number "10" (ACC ← 10h)
n+4	ADDAR 0x30,A ;Contents of the data memory address 30H and ACC are binary-added ;the result loads to the ACC (ACC ← A0h, C ← 0)
n+5	DAA ;Convert the content of ACC to decimal format, and restored to ACC ;The result in the ACC is "00" and the carry bit C is "1". This represents the ;decimal number "100"
n+6	...

DAS指令能在减法运算完成以后将ACC 里的数据从十六进制转化为十进制重存给ACC 转换操作在例子2.3中被说明

例2.3: DAS 转化

Address	Code
NA	#include <FC701.ASH>
n	...
n+1	MOVIA 0x10 ;Set immediate data = decimal format number "10" (ACC ← 10h)
n+2	MOVAR 0x30 ;Load immediate data "90" to data memory address 30H
n+3	MOVIA 0x20 ;Set immediate data = decimal format number "20" (ACC ← 20h)
n+4	SUBAR 0x30,A ;Contents of the data memory address 30H and ACC are binary-subtracted ;the result loads to the ACC (ACC ← F0h, C ← 0)
n+5	DAS ;Convert the content of ACC to decimal format, and restored to ACC ;The result in the ACC is "90" and the carry bit C is "0". This represents the ;decimal number "-10"
n+6	...



## 2.9 振荡器配置 (Oscillator Configurations)

SM701有七种不同的振荡模式，用户可通过编程Fosc配置位来选择相应的振荡方式：

- LF: 低频晶体谐振器
- HF: 高频晶体谐振器
- IRC:内部电阻内部电容振荡器
- ERIC:外部电阻内部电容振荡器
- ERC: 外部RC振荡器
- XT: 晶体/陶瓷振荡器
- IRC\_RTC: Fcpu使用内部电阻电容振荡器/T0使用低频晶体谐振器

在 LF, XT, HF 或 IRC\_RTC 模式下，一个晶体或陶瓷谐振器连接到 OSCI 和 OSCO 管脚建立振荡源。当在 LF, XT 或 HF 模式下，单片机通过 OSCI 脚接入外部时钟源。使用 ERC 振荡模式为成本节省主要使用在定时无须精确场合下的应用，RC 振荡器频率取决于电阻和电容 (Cext)，操作温度以及其他过程参数。

使用 IRC (IRC\_RTC) /ERIC 振荡模式为成本节省主要使用在定时无须精确场合下的应用，单片机具有 4 种不同的振荡频率，8MHz, 4MHz, 1MHz, 和 455KHz，通过 (RCM<1:0>)来选择一种，或则用户改变外部电阻来实现。ERIC 振荡器频率取决于电阻和电容 (Cext)，操作温度以及其他过程参数。

图2.16: HF, XT 和 LF 振荡器模式(晶振振荡器或陶瓷振荡器)

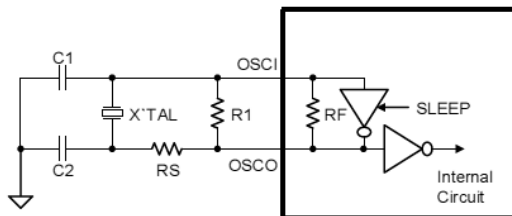


图2.17: HF, XT 和 LF 振荡器模式 (外部时钟输入操作)

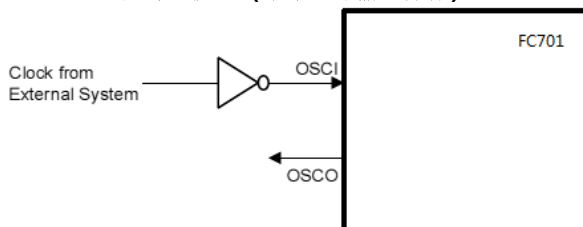


图2.18: ERC 振荡器模式 (外部电阻电容振荡器)

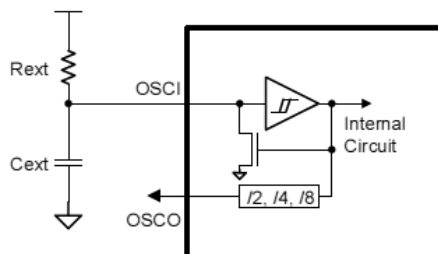




图2.19: ERIC 振荡器模式 (外部电阻 内部电容 振荡器)

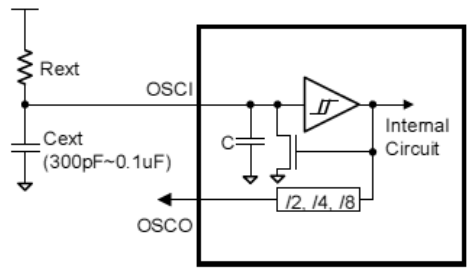
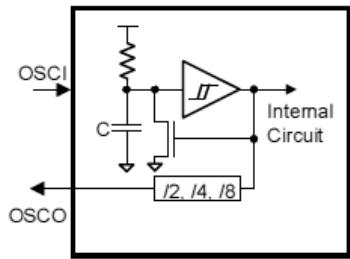


图2.20: IRC 振荡器模式 (内部电阻电容 振荡器)



## 2.10 配置选项

表2.4: 配置选项0

位	名称	说明
2~0	Fosc<2:0>	振荡源选择位 = 0 0 0→ERC mode (external R & C) (默认) IOB4/OSCO管脚为取OSCOUT功能 = 1 1 0→ HF mode = 1 0 1→XT mode = 1 0 0→LF mode = 0 1 1→IRC mode (internal R & C) IOB4/OSCO管脚为取OSCOUT功能 = 0 0 1→ERIC mode (external R & internal C) IOB4/OSCO管脚为取OSCOUT功能 = 1 1 1→IRC_RTC mode (external R & internal C for Fcpu/LF for T0)
5~3	LVDT<2:0>	低电压检测选择位 = 0 0 0→禁止低电压检测(默认) = 0 0 1→enable, LVDT voltage = 2.0V, 睡眠模式关闭 = 0 1 0→enable, LVDT voltage = 2.0V = 0 1 1→enable, LVDT voltage = 3.6V = 1 0 0→enable, LVDT voltage = 1.8V = 1 0 1→enable, LVDT voltage = 2.2V = 1 1 0→enable, LVDT voltage = 2.4V = 1 1 1→enable, LVDT voltage = 2.6V
7~6	RCM<1:0>	IRC选择位 = 0 0→4MHz (默认) = 0 1→8MHz = 1 0→1MHz = 1 1→455KHz
10~8	SUT<2:0>	PWRT & WDT计数周期选择位 = 0 0 0→PWRT = WDT prescaler rate = 18ms (默认) = 0 0 1→PWRT = WDT prescaler rate = 4.5ms = 0 1 0→PWRT = WDT prescaler rate = 288ms = 0 1 1→PWRT = WDT prescaler rate = 72ms = 1 0 0→PWRT = 640us, WDT prescaler rate = 18ms = 1 0 1→PWRT = 640us, WDT prescaler rate = 4.5ms = 1 1 0→PWRT = 640us, WDT prescaler rate = 288ms = 1 1 1→PWRT = 640us, WDT prescaler rate = 72ms
11	OSCOUT	IRC/ERIC/ERC 模式下IOB4/OSCO功能选择位置 = 0, IOB4(默认) = 1, OSCO
12	RSTBIN	IOB3/RSTB选择位置 = 0, IOB3 (默认) = 1, RSTB

表2.5: 配置选项1

位	名称	说明
0	WDTEN	看门狗使能位 = 0, 禁止WDT(默认) = 1, 使能WDT
1	PROTECT	代码保护选择位 = 0→代码不加密EPROM code protection off (默认) = 1→代码加密EPROM code protection on
3~2	OSCD<1:0>	指令运行周期选择位 = 0 0→4个振荡周期 (默认) = 1 0→2个振荡周期 = 1 1→8个振荡周期
4	PBWKEN	IOB口中断唤醒使能 = 0, IOB口中断唤醒使能通过WUCON寄存器控制, LVDTE软件控制 (默认) = 1, IOB口默认打开中断唤醒使能, LVDTE使能打开 (无法通过软件设置)
5	RDPORT	IO作为输出时, 读端口方式控制位 = 0, 从寄存器读 (默认) = 1, 从管脚读
6	SCHMITT	I/O输入缓冲控制位 = 0, 通过Schmitt触发器(默认) = 1, 不通过Schmitt触发器
7	IOB3OD-	IOB3开漏输出控制位 = 0, IOB3口为I/O口通过IOSTB3控制输入输出(默认) = 1, IOB3口作为输出时为开漏输出
12~8	OTPBANK<4:0>	OTP bank选择 (每个bank 256条指令, bank只能按bank0到bank3的顺序选择; 每个bank512条指令, bank只能按bank0到bank1的顺序选择) = 0 0 0 0 0, OTP 1K容量(默认) = 1 1 0 0 0, OTP 选择bank_256_0 (PC 000~0FF) = 1 1 0 0 1, OTP 选择bank_256_1 (PC 100~1FF) = 1 1 0 1 1, OTP 选择bank_256_2 (PC 200~2FF) = 1 1 1 1 1, OTP 选择bank_256_3 (PC 300~3FF) = 1 0 0 0 0, OTP 选择bank_512_0 (PC 000~1FF) = 1 0 1 0 0, OTP 选择bnak_512_1 (PC 200~3FF)

表2.6: 配置选项2

位	名称	说明
7~0	无定义	-
8	POWER_SAVE	455K节能选择位 = 0, IRC选择455K时降低功耗 (默认) = 1, 关闭455K节能模式
9	IO_DRIVE_WEAK	IO口弱驱动选择位 = 0, IO口标准驱动能力 (默认) = 1, IO口弱驱动能力
10	IO_DRIVE_STRONG	IO口强驱动选择位 = 0, IO口标准驱动能力 (默认) = 1, IO口强驱动能力
11	ERASE_TIME_DEC	内置E2P擦除时间选择位 = 0, 120ms左右(默认) = 1, 30ms左右

表2.7: Selection of IOB5/OSCI and IOB4/OSCO Pins

振荡方式	IOB5/OSCI	IOB4/OSCO
IRC/ERIC	IOB5 (OSCIN=0)	IOB4/OSCO selected by OSCOUT bit
	OSCI (OSCIN=1)	IOB4/OSCO selected by OSCOUT bit
ERC	OSCI	IOB4/OSCO selected by OSCOUT bit
HF	OSCI	OSCO
LF	OSCI	OSCO

### 3.0 指令集合

操作语法	说明	操作内容	指令周期	影响标志位
<b>BCR</b> R, bit	Clear bit in R	$0 \rightarrow R<b>$	1	-
<b>BSR</b> R, bit	Set bit in R	$1 \rightarrow R<b>$	1	-
<b>BTRSC</b> R, bit	Test bit in R, Skip if Clear	Skip if $R<b> = 0$	$1/2^{(1)}$	-
<b>BTRSS</b> R, bit	Test bit in R, Skip if Set	Skip if $R<b> = 1$	$1/2^{(1)}$	-
<b>NOP</b>	No Operation	No operation	1	-
<b>CLRWDT</b>	Clear Watchdog Timer	$00h \rightarrow WDT$ , $00h \rightarrow WDT$ prescaler	1	/TO, /PD
<b>OPTION</b>	Load OPTION register	$ACC \rightarrow OPTION$	1	-
<b>SLEEP</b>	Go into power-down mode	$00h \rightarrow WDT$ , $00h \rightarrow WDT$ prescaler	1	/TO, /PD
<b>DAA</b>	Adjust ACC's data format from HEX to DEC after any addition operation	$ACC(hex) \rightarrow ACC(dec)$	1	C
<b>DAS</b>	Adjust ACC's data format from HEX to DEC after any subtraction operation	$ACC(hex) \rightarrow ACC(dec)$	1	-
<b>RETURN</b>	Return from subroutine	Top of Stack $\rightarrow PC$	2	-
<b>RETFIE</b>	Return from interrupt, set GIE bit	Top of Stack $\rightarrow PC$ , $1 \rightarrow GIE$	2	-
<b>INT</b>	S/W interrupt	$PC + 1 \rightarrow$ Top of Stack, $002h \rightarrow PC$	2	-
<b>IOST</b> R	Load IOST register	$ACC \rightarrow IOST$ register	1	-
<b>CLRA</b>	Clear ACC	$00h \rightarrow ACC$	1	Z
<b>CLRR</b> R	Clear R	$00h \rightarrow R$	1	Z
<b>MOVAR</b> R	Move ACC to R	$ACC \rightarrow R$	1	-
<b>MOVR</b> R, d	Move R	$R \rightarrow dest$	1	Z
<b>DECR</b> R, d	Decrement R	$R - 1 \rightarrow dest$	1	Z
<b>DECRSZ</b> R, d	Decrement R, Skip if 0	$R - 1 \rightarrow dest$ , Skip if result = 0	$1/2^{(1)}$	-
<b>INCR</b> R, d	Increment R	$R + 1 \rightarrow dest$	1	Z
<b>INCRSZ</b> R, d	Increment R, Skip if 0	$R + 1 \rightarrow dest$ , Skip if result = 0	$1/2^{(1)}$	-
<b>ADDAR</b> R, d	Add ACC and R	$R + ACC \rightarrow dest$	1	C, DC, Z
<b>SUBAR</b> R, d	Subtract ACC from R	$R - ACC \rightarrow dest$	1	C, DC, Z
<b>ADCAR</b> R, d	Add ACC and R with Carry	$R + ACC + C \rightarrow dest$	1	C, DC, Z
<b>SBCAR</b> R, d	Subtract ACC from R with Carry	$R + ACC + C \rightarrow dest$	1	C, DC, Z
<b>ANDAR</b> R, d	AND ACC with R	$ACC$ and $R \rightarrow dest$	1	Z
<b>IORAR</b> R, d	Inclusive OR ACC with R	$ACC$ or $R \rightarrow dest$	1	Z
<b>XORAR</b> R, d	Exclusive OR ACC with R	$R$ xor $ACC \rightarrow dest$	1	Z
<b>COMR</b> R, d	Complement R	$R \rightarrow dest$	1	Z
<b>RLR</b> R, d	Rotate left f through Carry	$R<7> \rightarrow C$ , $R<6:0> \rightarrow dest<7:1>$ , $C \rightarrow dest<0>$	1	C
<b>RRR</b> R, d	Rotate right f through Carry	$C \leftarrow dest<7>$ , $R<7:1> \leftarrow dest<6:0>$ , $R<0> \leftarrow C$	1	C
<b>SWAPR</b> R, d	Swap R	$R<3:0> \rightarrow dest<7:4>$ , $R<7:4> \rightarrow dest<3:0>$	1	-
<b>MOVIA</b> I	Move Immediate to ACC	$I \rightarrow ACC$	1	-
<b>ADDIA</b> I	Add ACC and Immediate	$I + ACC \rightarrow ACC$	1	C, DC, Z
<b>SUBIA</b> I	Subtract ACC from Immediate	$I - ACC \rightarrow ACC$	1	C, DC, Z
<b>ANDIA</b> I	AND Immediate with ACC	$ACC$ and $I \rightarrow ACC$	1	Z
<b>IORIA</b> I	OR Immediate with ACC	$ACC$ or $I \rightarrow ACC$	1	Z

操作语法	说明	操作内容	指令周期	影响标志位
<b>XORIA</b> I	Exclusive OR Immediate to ACC	ACC xor I → ACC	1	Z
<b>RETIA</b> I	Return, place Immediate in ACC	I ACC, Top of Stack → PC	2	-
<b>CALL</b> I	Call subroutine	PC + 1 → Top of Stack, I → PC	2	-
<b>GOTO</b> I	Unconditional branch	I → PC	2	-

注释: 1. 两周期指令为分支跳转指令

2. bit : Bit 地址为8位寄存器R中的某一位

R : 寄存器地址 (00h to 3Fh)

I : 立即数

ACC : 累加器

d : 目的选择:

=0 (结果存放在ACC)

=1 (结果存放在R)

dest : 目的地

PC : 程序指针

PCHBUF : 高位缓冲程序指针

WDT : 看门狗计数器

GIE : 中断允许总控制位

TO : 计数溢出位

PD : 省电模式选择位

C : 进位/借位标志

DC : 辅助进位/借位标志.(低四位向高四位进位/借位标志)

Z : 零标志

<b>ADCAR(带进位加法)</b>	<b>Add ACC and R with Carry</b>
语法	ADCAR R, d
操作数	$0 \leq R \leq 63$ $d \in [0, 1]$
操作内容	$R + ACC + C \rightarrow dest$
受影响的标志	C, DC, Z
说明	将A寄存器的内含值加上R寄存器的内含值（带进位），如果d是0结果在ACC中存放。如果d是1结果在R中存放。
指令执行周期	1
<b>ADDAR (加法指令)</b>	<b>ACC and R with Carry</b>
语法	ADDAR R, d
操作数	$0 \leq R \leq 63$ $d \in [0, 1]$
操作内容	$R + ACC \rightarrow dest$
受影响的标志	C, DC, Z
说明	将A寄存器的内含值加上R寄存器的内含值（不带进位），如果d是0结果在ACC中存放。如果d是1结果在R中存放。
指令执行周期	1
<b>ADDIA</b>	<b>Add ACC and Immediate</b>
语法	ADDIA I
操作数	$0 \leq I \leq 255$
操作内容	$ACC + I \rightarrow ACC$
受影响的标志	C, DC, Z
说明	将A寄存器的内含值加上立即数I，结果在ACC中存放。
指令执行周期	1
<b>ANDAR</b>	<b>AND ACC and R</b>
语法	ANDAR R, d
操作数	$0 \leq R \leq 63$ $d \in [0, 1]$
操作内容	$ACC \text{ and } R \rightarrow dest$
受影响的标志	Z
说明	将A寄存器内含值和R寄存器做相与操作，如果d是0结果在ACC中存放。如果d是1结果在R中存放。
指令执行周期	1
<b>ANDIA</b>	<b>AND Immediate with ACC</b>
语法	ANDIA I
操作数	$0 \leq I \leq 255$
操作内容	$ACC \text{ and } I \rightarrow dest$
受影响的标志	Z
说明	将A寄存器的内含值与立即数I做相与操作，结果在ACC中存放
指令执行周期	1
<b>BCR</b>	<b>Clear Bit in R</b>
语法	BCR R, b
操作数	$0 \leq R \leq 63$ $0 \leq b \leq 7$
操作内容	$0 \rightarrow R<b>$
受影响的标志	无
说明	R寄存器的位“b”被设为0
指令执行周期	1
<b>BSR</b>	<b>Set Bit in R</b>

语 法 BSR R, b  
操作数  $0 \leq R \leq 63$   
 $0 \leq b \leq 7$   
操作内容  $1 \rightarrow R<b>$   
受影响的标志 无  
说 明 R寄存器的位“b”被设成1  
指令执行周期 1

**BTRSC Test Bit in R, Skip if Clear**  
语 法 BTRSC R, b  
操作数  $0 \leq R \leq 63$   
 $0 \leq b \leq 7$   
操作内容 当 $R<b> = 0$ 跳过后条指令  
受影响的标志 无  
说 明  $R<b> = 0$ 跳过后条指令  
 $R<b> = 0$ 时, 该指令周期中提取的下条指令被丢弃, 并以执行NOP  
操作来替换这条2周期指令。  
指令执行周期 1(2)

**BTRSS Test Bit in R, Skip if Set**  
语 法 BTRSS R, b  
操作数  $0 \leq R \leq 63$   
 $0 \leq b \leq 7$   
操作内容 当 $R<b> = 1$ 跳过后条指令  
受影响的标志 无  
说 明  $R<b> = 1$ 跳过后条指令  
 $R<b> = 1$ 时, 该指令周期中提取的下条指令被丢弃, 并以执行NOP  
操作来替换这条2周期指令。  
指令执行周期 1(2)

**CALL Subroutine Call**  
语 法 CALL I  
操作数  $0 \leq I \leq 1023$   
操作内容  $PC + 1 \rightarrow \text{Top of Stack}$   
 $I \rightarrow PC$   
受影响的标志 无  
说 明 子程序调用。首先下一条指令地址(PC+1)进栈。10位  
立即地址被装载入PC指针的位<9:0>。CALL是二周期指令。  
指令执行周期 2

**CLRA Clear ACC**  
语 法 CLRA  
操作数 无  
操作内容  $00h \rightarrow \text{ACC}$   
 $I \rightarrow Z$   
受影响的标志 Z  
说 明 ACC被清零, Z标志为置1  
指令执行周期 1

**CLRR Clear R**  
语 法 CLRR R  
操作数  $0 \leq R \leq 63$   
操作内容  $00h \rightarrow R$   
 $I \rightarrow Z$   
受影响的标志 Z  
说 明 R被清零, Z标志为置1  
指令执行周期 1

**CLRWDT Clear Watchdog Timer**



语 法 CLRWDT  
操作数 无  
操作内容 00h → WDT:  
00h → WDT prescaler (已经设置了WDT预置器):  
1 → TO:  
1 → PD  
受影响的标志 TO, PD  
说明 CLRWDT指令重置WDT, 如已经设置了WDT预置器, 也重置WDT预置器; 并把TO, PD位置1  
指令执行周期 1

## COMR

## Complement R

语 法 COMR R, d  
操作数  $0 \leq R \leq 63$   
 $d \in [0, 1]$   
操作内容  $R \rightarrow \text{dest}$   
受影响的标志 Z  
说明 将R内含内容取反, 如果d是0结果在ACC中存放。 如果d是1结果在R中存放。  
指令执行周期 1

## DAA

## Adjust ACC's data format from HEX to DEC

语 法 DAA  
操作数 无  
操作内容  $\text{ACC(hex)} \rightarrow \text{ACC(dec)}$   
受影响的标志 C  
说明 在有些加法操作以后把ACC内值的十六进制转化十进制,  
指令执行周期 1

## DAS

## Adjust ACC's data format from HEX to DEC

语 法 DAS  
操作数 无  
操作内容  $\text{ACC(hex)} \rightarrow \text{ACC(dec)}$   
受影响的标志 C  
说明 在有些减法操作以后把ACC内值的十六进制转化十进制,  
指令执行周期 1

## DECR

## Decrement R

语 法 DECR R, d  
操作数  $0 \leq R \leq 63$   
 $d \in [0, 1]$   
操作内容  $R - 1 \rightarrow \text{dest}$   
受影响的标志 Z  
说明 递减R寄存器的值, 如果d是0结果在ACC中存放。 如果d是1结果在R中存放。  
指令执行周期 1

## DECRSZ

## Decrement R, Skip if 0

语 法 DECRSZ R, d  
操作数  $0 \leq R \leq 63$   
 $d \in [0, 1]$   
操作内容  $R - 1 \rightarrow \text{dest}$  如果结果等于0, 跳过下条指令  
受影响的标志 无  
说明 递减R寄存器的值, 如果d是0结果在ACC中存放。 如果d是1结果在R中存放。  
如果结果等于0, 该指令周期中提取的下条指令被丢弃, 并以执行NOP操作来替换这条2周期指令。  
指令执行周期 1(2)

## GOTO

## Unconditional Branch

语法	GOTO I
操作数	$0 \leq I \leq 1023$
操作内容	$I \rightarrow PC$
受影响的标志	无
说明	无条件跳转。10位立即地址被装载入PC指针的位<9:0>。GOTO是二周期指令。
指令执行周期	2

### INCR

### Increment R

语法	INCR R, d
操作数	$0 \leq R \leq 63$ $d \in [0, 1]$
操作内容	$R + 1 \rightarrow dest$
受影响的标志	Z
说明	将被指定R寄存器的内含值加1, 如果d是0结果在ACC中存放。 <sup>d</sup> 如果是1结果在R中存放。
指令执行周期	1

### INCRSZ

### Increment R, Skip if 0

语法	INCRSZ R, d
操作数	$0 \leq R \leq 63$ $d \in [0, 1]$
操作内容	$R + 1 \rightarrow dest$ 如果结果等于0, 跳过下条指令
受影响的标志	无
说明	将被指定R寄存器的内含值加1, 如果d是0结果在ACC中存放。 如果d是1结果在R中存放。 如果结果等于0, 该指令周期中提取的下条指令被丢弃, 并以执行NOP操作来替换这条2周期指令。
指令执行周期	1(2)

### INT

### S/W Interrupt

语法	INT
操作数	无
操作内容	$PC + 1 \rightarrow Top\ of\ Stack:$ $002h \rightarrow PC$
受影响的标志	无
说明	子程序调用。 首先下一条指令地址(PC+1)进栈。 10位地址002h被装载入PC指针的位<9:0>。CALL是二周期指令。
指令执行周期	2

### IORAR

### OR ACC with R

语法	IORAR
操作数	$0 \leq R \leq 63$ $d \in [0, 1]$
操作内容	$ACC\ or\ R \rightarrow dest$
受影响的标志	Z
说明	将A寄存器内含值和R寄存器做或操作, 如果d是0结果在ACC中存放。 如果d是1结果在R中存放。
指令执行周期	1

### IORIA

### OR Immediate with ACC

语法	IORIA I
操作数	$0 \leq I \leq 255$
操作内容	$ACC\ or\ I \rightarrow dest$
受影响的标志	Z
说明	将A寄存器的内含值与立即数I做相与操作, 结果在ACC中存放
指令执行周期	1

### IOST

### Load IOST Register

语 法 IOST R  
操作数 R = 5 or 6  
操作内容 ACC → IOST register R  
受影响的标志 无  
说 明 将A寄存器的内含值加载到IOST register R中  
指令执行周期 1

---

**MOVAR Move ACC to R**

语 法 MOVAR R  
操作数  $0 \leq R \leq 63$   
操作内容 ACC → R  
受影响的标志 无  
说 明 将数据从ACC传送到R  
指令执行周期 1

---

**MOVIA Move Immediate to ACC**

语 法 MOVIA I  
操作数  $0 \leq I \leq 255$   
操作内容 I → ACC  
受影响的标志 无  
说 明 将立即值载入A寄存器中  
指令执行周期 1

---

**MOVR Move Immediate to ACC**

语 法 MOVR R, d  
操作数  $0 \leq R \leq 63$   
 $d \in [0, 1]$   
操作内容 R → dest  
受影响的标志 无 d是1  
说 明 将R寄存器内容载入R中，如果d是0结果在ACC中存放。如果结果在R中存放。d 为1用来测试该寄存器对标志Z 是否有影响  
指令执行周期 1

---

**NOP No Operation**

语 法 NOP  
操作数 无  
操作内容 无操作  
受影响的标志 无  
说 明 不做任何操作  
指令执行周期 1

---

**OPTION Load OPTION Register**

语 法 OPTION  
操作数 无  
操作内容 ACC → OPTION  
受影响的标志 无  
说 明 将A寄存器内容载入OPTION中  
指令执行周期 1

---

**RETFIE Return from Interrupt, Set 'GIE' Bit**

语 法 RETFIE  
操作数 无  
操作内容 Top of Stack → PC  
受影响的标志 无  
说 明 程序计数器载入堆栈返回地址。 GIE位被设置到1。 这是二周期指令。  
指令执行周期 2

---

**RETIA Return with Immediate in ACC**

语 法	RETIA I
操作数	$0 \leq I \leq 255$
操作内容	I → ACC: Top of Stack → PC
受影响的标志	无
说 明	程序计数器载入堆栈返回地址，并把立即数送入A中。这是二周期指令。
指令执行周期	2

### RETURN

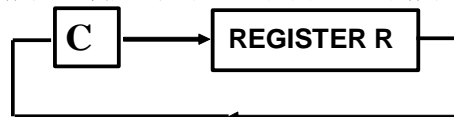
### Return from Subroutine

语 法	RETURN
操作数	无
操作内容	Top of Stack → PC
受影响的标志	无
说 明	程序计数器载入堆栈返回地址。这是二周期指令。
指令执行周期	2

### RLR

### Rotate Left f through Carry

语 法	RLR R, d
操作数	$0 \leq R \leq 63$ $d \in [0, 1]$
操作内容	$R \langle 7 \rangle \rightarrow C$ : $R \langle 6:0 \rangle \rightarrow \text{dest} \langle 7:1 \rangle$ : $C \rightarrow \text{dest} \langle 0 \rangle$
受影响的标志	C
说 明	R寄存器的内含值又移1bit，右移时包含C(进位标志)，如下图，结果存放由d决定，如果d是0结果在ACC中存放。如果d是1结果在R中存放。



指令执行周期	1
--------	---

### RRR

### Rotate Right f through Carry

操作数	$0 \leq R \leq 63$
操作内容	$C \rightarrow \text{dest} \langle 7 \rangle$ : $R \langle 7:1 \rangle \rightarrow \text{dest} \langle 6:0 \rangle$ :
说 明	R寄存器的内含值又移1bit，右移时包含C(进位标志)，如下图，结果由d决定，如果d是0结果在ACC中存放。如果d是1结果在R中存放。



指令执行周期	1
--------	---

### SLEEP

### SLEEP

操作内容	00h → WDT: 00h → WDT prescaler: 1 → TO:
说 明	TO位置1。PD位清零，WDT和WDT预置器清零
指令执行周期	1

### SBCAR (带借位加法)

### Subtract ACC from R with Carry

语法	SBCAR R, d
操作数	$0 \leq R \leq 63$ $d \in [0, 1]$
操作内容	$(R - ACC - C) \rightarrow dest$
受影响的标志	C, DC, Z
说明	将R寄存器的内含值减去A寄存器的内含值（带借位），如果d是0结果在ACC中存放。如果d是1结果在R中存放。
指令执行周期	1
<b>SUBAR Subtract ACC from R</b>	
语法	SUBAR R, d
操作数	$0 \leq R \leq 63$ $d \in [0, 1]$
操作内容	$R - ACC \rightarrow dest$
受影响的标志	C, DC, Z
说明	将R寄存器的内含值减去A寄存器的内含值（不带借位），如果d是0结果在ACC中存放。如果d是1结果在R中存放。
指令执行周期	1
<b>SUBIA Subtract ACC from Immediate</b>	
语法	SUBIA I
操作数	$0 \leq I \leq 255$
操作内容	$ACC - I \rightarrow ACC$
受影响的标志	C, DC, Z
说明	将A寄存器的内含值减去立即数I，结果在ACC中存放。
指令执行周期	1
<b>SWAPR Swap nibbles in R</b>	
语法	SWAPR R, d
操作数	$0 \leq R \leq 63$ $d \in [0, 1]$
操作内容	$R<3:0> \rightarrow dest<7:4>$ $R<7:4> \rightarrow dest<3:0>$
受影响的标志	无
说明	将所选定的寄存器，高4位以及低4位，如果d是0结果在ACC中存放。如果d是1结果在R中存放。
指令执行周期	1
<b>XORAR Exclusive OR ACC with R</b>	
语法	XORAR R, d
操作数	$0 \leq R \leq 63$ $d \in [0, 1]$
操作内容	$ACC \text{ xor } R \rightarrow dest$
受影响的标志	Z
说明	将A寄存器的值和R寄存器的值异或在一起，如果d是0结果在ACC中存放。如果d是1结果在R中存放。
指令执行周期	1
<b>XORIA Exclusive OR Immediate with ACC</b>	
语法	XORIA I
操作数	$0 \leq I \leq 255$
操作内容	$ACC \text{ xor } I \rightarrow ACC$
受影响的标志	Z
说明	将A寄存器的值和立即数I异或在一起，结果在ACC中存放。
指令执行周期	1

## 4.0 绝对最大额定值

操作环境温度	0°C到+70°C
存储器额定温度	-65°C到+150°C
DC 电源电压 (Vdd)	0V 到+6.0V
输入电压 (对地电压 (Vss))	-0.3V 到(Vdd + 0.3)V

## 5.0 操作条件

DC 供电电压	+2.0V到+5.5V
操作温度	0°C到+70°C

\*细节详见 6.1

## 6.0 电气特性

### 6.1 SM701 电气特性

电气特性是在四时钟指令周期和 WDT & LVDT 禁用情况下

Ta=25°C

Sym	Description	Conditions	Min.	Typ.	Max.	Unit	
V <sub>DD</sub>	Supply voltage	0Hz~1MHz		2.0	5.5	V	
		1MHz~4MHz		2.0	5.5		
		4MHz~8MHz		2.4	5.5		
		8MHz~10MHz		2.6	5.5		
		10MHz~16MHz		3.0	5.5		
		16MHz~20MHz		3.5	5.5		
T <sub>PWR</sub>	Power rising time	V <sub>dd</sub> =0V to V <sub>dd</sub>	0.8		2.6	ms/V	
F <sub>HF</sub>	X'tal oscillation range	HF mode, V <sub>dd</sub> =5V	4		20	MHz	
		HF mode, V <sub>dd</sub> =3V	4		16		
F <sub>XT</sub>	X'tal oscillation range	XT mode, V <sub>dd</sub> =5V	0.455		20	MHz	
		XT mode, V <sub>dd</sub> =3V	0.455		8		
F <sub>LF</sub>	X'tal oscillation range	LF mode, V <sub>dd</sub> =5V	32		455	MHz	
		LF mode, V <sub>dd</sub> =3V	32		455		
F <sub>ERC</sub>	RC oscillation range	ERC mode, V <sub>dd</sub> =5V	DC		16	MHz	
		ERC mode, V <sub>dd</sub> =3V	DC		13		
F <sub>IRC/ERIC</sub>	RC oscillation range	ERIC mode, external R, V <sub>dd</sub> =5V	DC		16	MHz	
		ERIC mode, external R, V <sub>dd</sub> =3V	DC		16		
		IRC mode, internal R, V <sub>dd</sub> =5V	0.455		8		
		IRC mode, internal R, V <sub>dd</sub> =3V	0.455		8		
V <sub>IH</sub>	Input high voltage	With Schmitt-trigger					V
		I/O ports, V <sub>dd</sub> =5V	2.2		V <sub>DD</sub>		
		RSTB, T0CKI pins, V <sub>dd</sub> =5V	2.2		V <sub>DD</sub>		
		I/O ports, V <sub>dd</sub> =3V	1.7		V <sub>DD</sub>		
		RSTB, T0CKI pins, V <sub>dd</sub> =3V	1.7		V <sub>DD</sub>		
		Without Schmitt-trigger					
		I/O ports, V <sub>dd</sub> =5V	2.0		V <sub>DD</sub>		
		RSTB, T0CKI pins, V <sub>dd</sub> =5V	2.0		V <sub>DD</sub>		
		I/O ports, V <sub>dd</sub> =3V	1.5		V <sub>DD</sub>		
		RSTB, T0CKI pins, V <sub>dd</sub> =3V	1.5		V <sub>DD</sub>		
V <sub>IL</sub>	Input low voltage	With Schmitt-trigger					V
		I/O ports, V <sub>dd</sub> =5V	V <sub>SS</sub>		0.8		
		RSTB, T0CKI pins, V <sub>dd</sub> =5V	V <sub>SS</sub>		0.8		
		I/O ports, V <sub>dd</sub> =3V	V <sub>SS</sub>		0.5		
		RSTB, T0CKI pins, V <sub>dd</sub> =3V	V <sub>SS</sub>		0.5		
		Without Schmitt-trigger					
		I/O ports, V <sub>dd</sub> =5V	V <sub>SS</sub>		0.8		
		RSTB, T0CKI pins, V <sub>dd</sub> =5V	V <sub>SS</sub>		0.8		
		I/O ports, V <sub>dd</sub> =3V	V <sub>SS</sub>		0.6		
		RSTB, T0CKI pins, V <sub>dd</sub> =3V	V <sub>SS</sub>		0.6		

Sym	Description	Conditions	Min.	Typ.	Max.	Unit	
V <sub>OH</sub>	Output high voltage	I <sub>OH</sub> =-5.4mA, V <sub>DD</sub> =5V	3.6			V	
V <sub>OL</sub>	Output low voltage	I <sub>OL</sub> =8.7mA, V <sub>DD</sub> =5V			0.6	V	
I <sub>PH</sub>	Pull-high current	Input pin at V <sub>SS</sub> , V <sub>DD</sub> =5V	-50	-66	-80	uA	
I <sub>PL</sub>	Pull-down current	Input pin at V <sub>DD</sub> , V <sub>DD</sub> =5V	30	44	60	uA	
I <sub>WDT</sub>	WDT current (18mS)	V <sub>DD</sub> =5V		6		uA	
		V <sub>DD</sub> =3V		1.5			
T <sub>WDT</sub>	WDT period (18mS)	V <sub>DD</sub> =3V		20.2		mS	
		V <sub>DD</sub> =4V		17.5			
		V <sub>DD</sub> =5V		15.9			
I <sub>LVDT</sub>	LVDT current	V <sub>DD</sub> =5V, LVDT=3.6V		1.89		uA	
		V <sub>DD</sub> =5V, LVDT=2.4V		2.4			
		V <sub>DD</sub> =3V, LVDT=2.4V		1			
I <sub>SB</sub>	Power down current	Sleep mode, V <sub>DD</sub> =5V, WDT LVDT disable		1	2	uA	
		Sleep mode, V <sub>DD</sub> =3V, WDT LVDT disable		0.5	1		
I <sub>DD</sub>	Operating current	HF mode, V <sub>DD</sub> =5V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF					mA
		20MHz				3.03	
		16MHz				2.61	
		HF mode, V <sub>DD</sub> =3V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF					
		20MHz				1.27	
		16MHz				1.07	
I <sub>DD</sub>	Operating current	HF mode, V <sub>DD</sub> =5V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF					mA
		20MHz				4.36	
		16MHz				3.41	
		HF mode, V <sub>DD</sub> =3V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF					
		20MHz				1.94	
		16MHz				1.46	
I <sub>DD</sub>	Operating current	XT mode, V <sub>DD</sub> =5V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF					mA
		20MHz				2.13	
		16MHz				1.77	
		XT mode, V <sub>DD</sub> =5V, 4 clock instruction, OSCI / OSCO = 20pF / 100pF					
		455KHz				0.42	
		32KHz				-	
I <sub>DD</sub>	Operating current	XT mode, V <sub>DD</sub> =3V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF					mA
		16MHz				0.76	
		8MHz				0.43	
		XT mode, V <sub>DD</sub> =3V, 4 clock instruction, OSCI / OSCO = 20pF / 100pF					
		455KHz				0.16	
		32KHz				0.10	
I <sub>DD</sub>	Operating current	XT mode, V <sub>DD</sub> =5V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF					mA
		20MHz				3.12	
		16MHz				2.57	
		XT mode, V <sub>DD</sub> =5V, 2 clock instruction, OSCI / OSCO = 20pF / 100pF					
		455KHz				0.44	

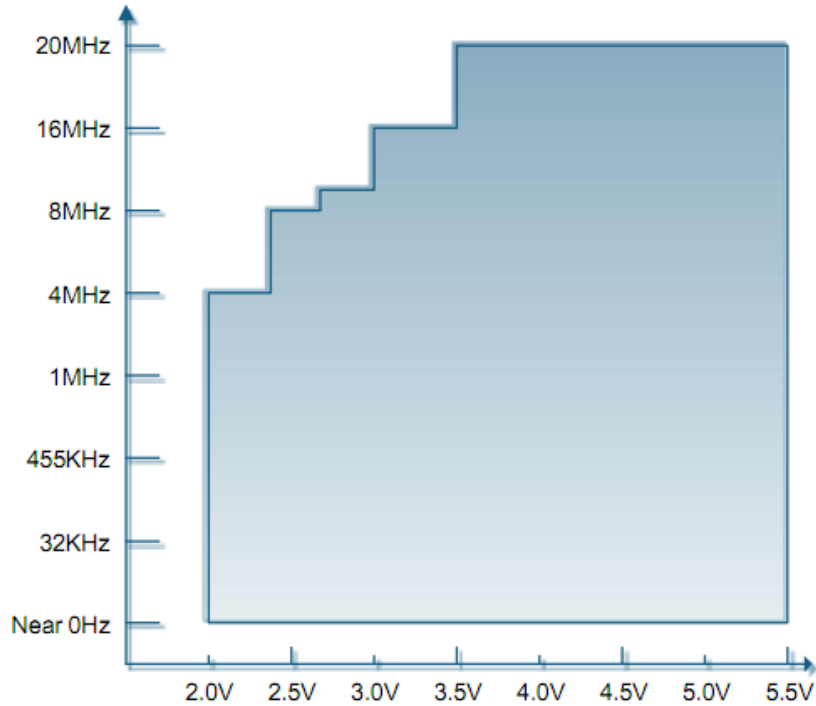


Sym	Description	Conditions	Min.	Typ.	Max.	Unit		
		32KHz		-				
I <sub>DD</sub>	Operating current	XT mode, V <sub>dd</sub> =3V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF				mA		
		16MHz		-				
		8MHz		0.62				
		XT mode, V <sub>dd</sub> =3V, 2 clock instruction, OSCI / OSCO = 20pF / 100pF						
		455KHz		0.44				
		32KHz		-				
I <sub>DD</sub>	Operating current	LF mode, V <sub>dd</sub> =5V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF				uA		
		455KHz		106.3				
		32KHz		42.5				
		LF mode, V <sub>dd</sub> =3V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF						
		455KHz		38.7				
		32KHz		12.1				
I <sub>DD</sub>	Operating current	LF mode, V <sub>dd</sub> =5V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF				uA		
		455KHz		130.9				
		32KHz		44.8				
		LF mode, V <sub>dd</sub> =3V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF						
		455KHz		50.4				
		32KHz		13				
I <sub>DD</sub>	Operating current	ERC mode, V <sub>dd</sub> =5V, 4 clock instruction				mA		
		C=3P	R=10Kohm	F=7.4MHz			1.2	
			R=3.3Kohm	F=16.4MHz			2.87	
		ERC mode, V <sub>dd</sub> =3V, 4 clock instruction						
		C=3P	R=10Kohm	F=7MHz			0.61	
			R=3.3Kohm	F=13.6MHz			1.41	
I <sub>DD</sub>	Operating current	ERIC mode, external R, V <sub>dd</sub> =5V, 4 clock instruction				mA		
		C=3P	R=10Kohm	F=7.6MHz			1.87	
			R=3.3Kohm	F=15.1MHz			3.94	
		ERIC mode, external R, V <sub>dd</sub> =3V, 4 clock instruction						
		C=3P	R=10Kohm	F=6.7MHz			0.87	
			R=3.3Kohm	F=13.7MHz			2	
I <sub>DD</sub>	Operating current	ERIC mode, external R, V <sub>dd</sub> =5V, 2 clock instruction				mA		
		F=8MHz	R=103.2Kohm		1.11			
		F=4MHz	R=206.2Kohm		0.58			
		ERIC mode, external R, V <sub>dd</sub> =3V, 4 clock instruction						
		F=8MHz	R=99.4Kohm		0.56			
		F=4MHz	R=187.6Kohm		0.29			
I <sub>DD</sub>	Operating current	ERIC mode, external R, V <sub>dd</sub> =5V, 2 clock instruction				mA		
		F=8MHz	R=103.2Kohm		1.78			
		F=4MHz	R=206.2Kohm		0.9			
		ERIC mode, external R, V <sub>dd</sub> =3V, 2 clock instruction						
		F=8MHz	R=99.4Kohm		0.89			
		F=4MHz	R=187.6Kohm		0.46			

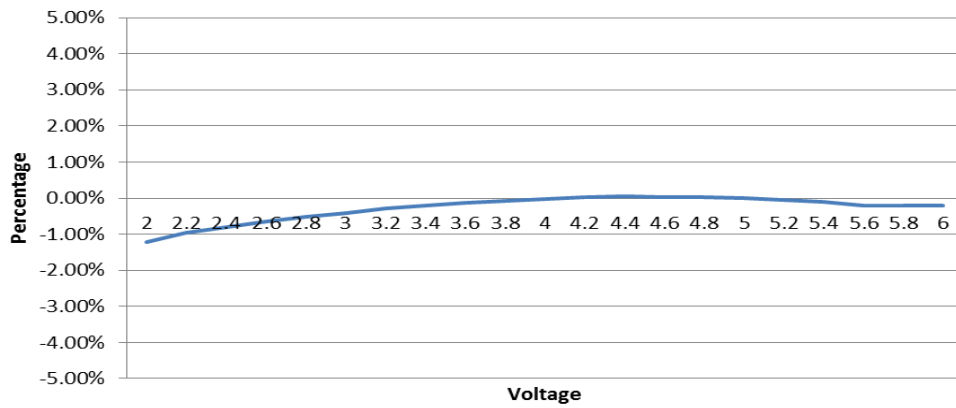
Sym	Description	Conditions	Min.	Typ.	Max.	Unit
I <sub>DD</sub>	Operating current	IRC mode, internal R, V <sub>dd</sub> =5V, 4 clock instruction				mA
		F=8MHz		1.77		
		F=4MHz		0.93		
		F=1MHz		0.32		
		F=455KHz		0.21		
		IRC mode, internal R, V <sub>dd</sub> =3V, 4 clock instruction				
		F=8MHz		0.97		
		F=4MHz		0.55		
		F=1MHz		0.2		
		F=455KHz		0.14		
I <sub>DD</sub>	Operating current	IRC mode, internal R, V <sub>dd</sub> =5V, 2 clock instruction				mA
		F=8MHz		1.75		
		F=4MHz		0.95		
		F=1MHz		0.33		
		F=455KHz		0.22		
		IRC mode, internal R, V <sub>dd</sub> =3V, 2 clock instruction				
		F=8MHz		0.99		
		F=4MHz		0.55		
		F=1MHz		0.21		
		F=455KHz		0.15		
I <sub>ER</sub>	EEPROM erase current	V <sub>dd</sub> =5V		10		mA
I <sub>PG</sub>	EEPROM prog current	V <sub>dd</sub> =5V		5		mA

## 6.2 SM701电气特性表

### 6.2.1 操作频率 vs 操作电压 (Ta=25°C)

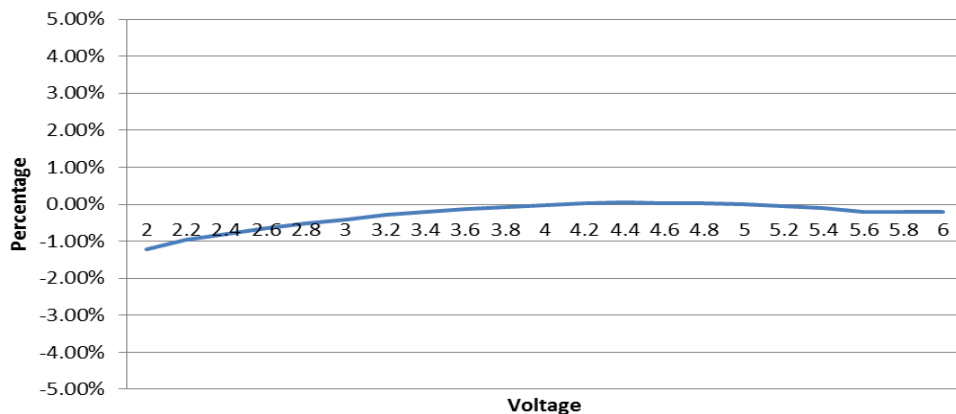


### 6.2.2 内部 4MHz RC vs 供应电压 (Ta=25°C)



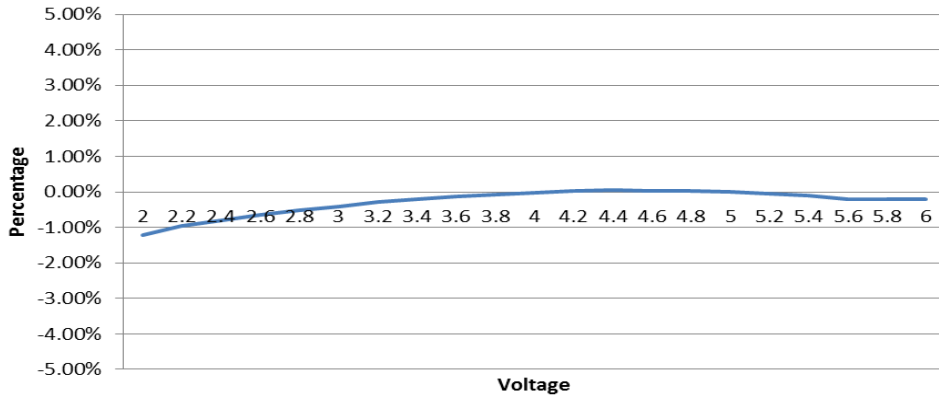
注: 曲线仅供设计参考

### 6.2.3 内部 8MHz RC vs 供应电压 (Ta=25°C)



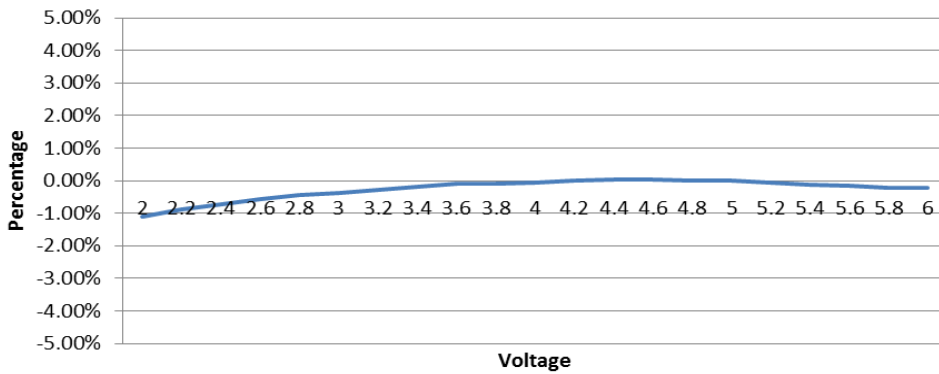
注: 曲线仅供设计参考

### 6.2.4 内部 1MHz RC vs 供应电压 (Ta=25°C)



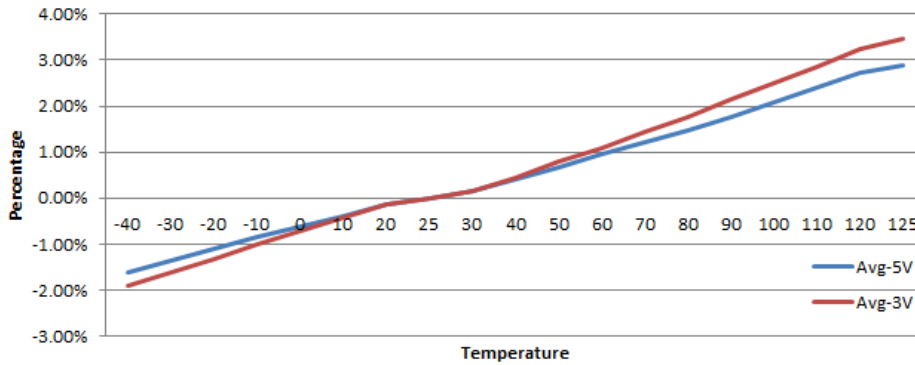
注: 曲线仅供设计参考

### 6.2.5 内部 455KHz RC vs 供应电压 (Ta=25°C)



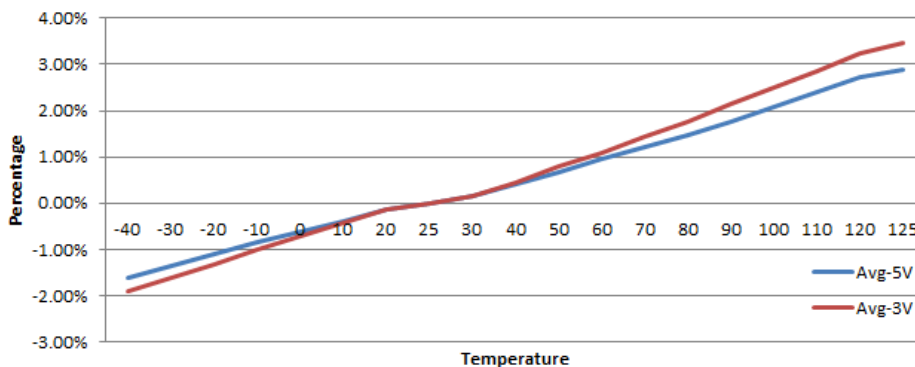
注: 曲线仅供设计参考

### 6.2.6 内部 4MHz RC vs 温度



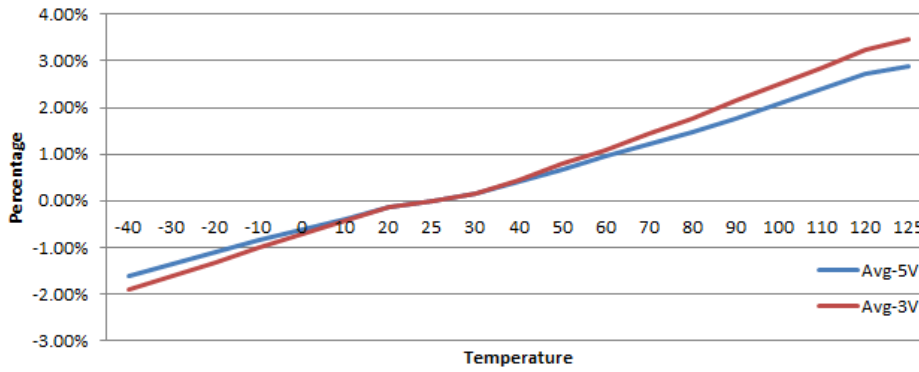
注: 曲线仅供设计参考

### 6.2.7 内部 8MHz RC vs 温度



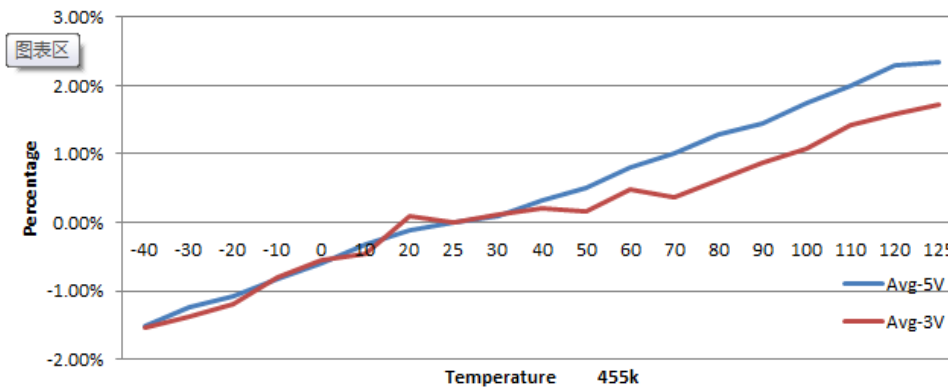
注: 曲线仅供设计参考

### 6.2.8 内部 1MHz RC vs 温度



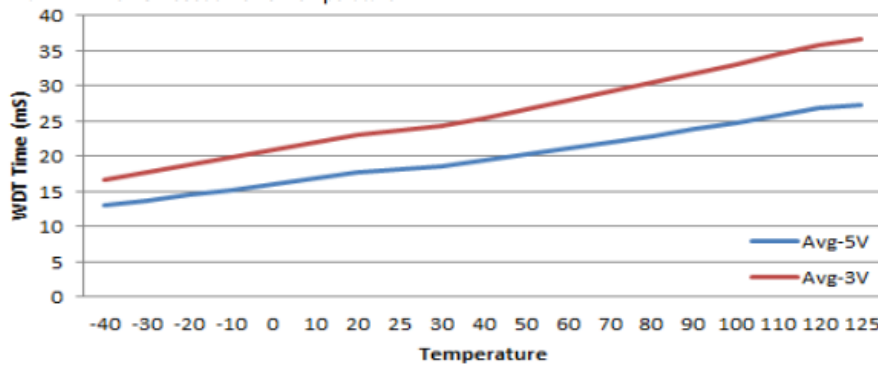
注: 曲线仅供设计参考

### 6.2.9 内部 455KH Hz RC vs 温度



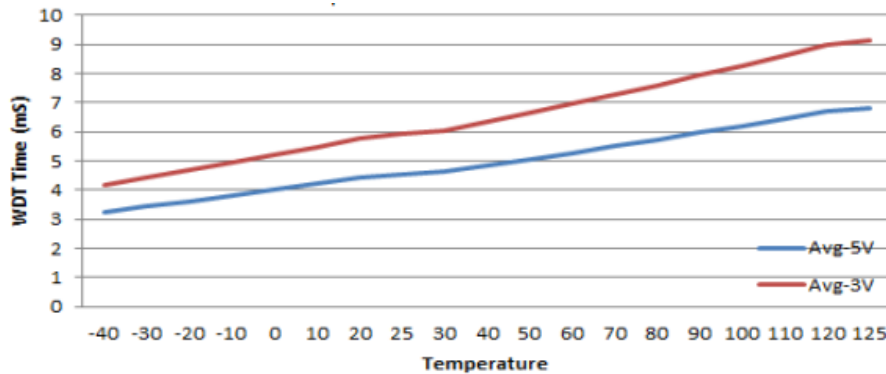
注: 曲线仅供设计参考

### 6.2.10 WTD18 毫秒复位时间 vs 温度



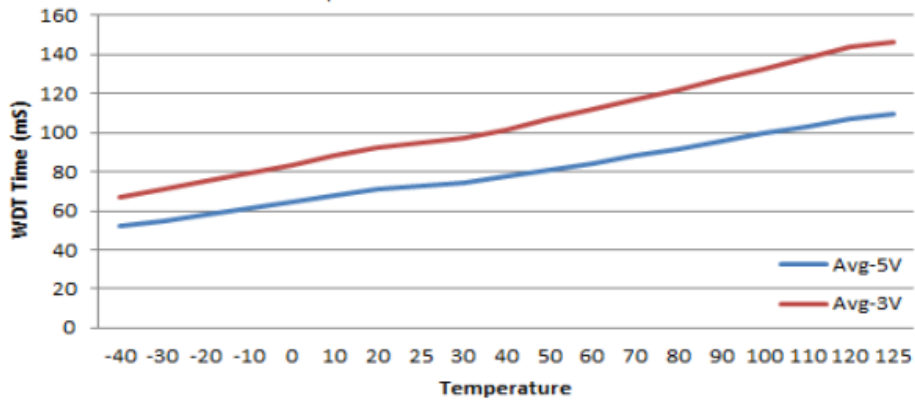
注: 曲线仅供设计参考

### 6.2.11 WTD4.5 毫秒复位时间 vs 温度



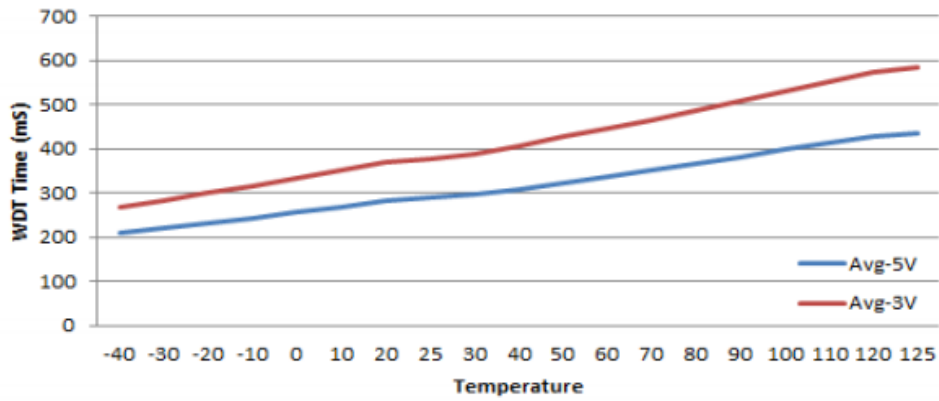
注: 曲线仅供设计参考

### 6.2.12 WTD72 毫秒复位时间 vs 温度



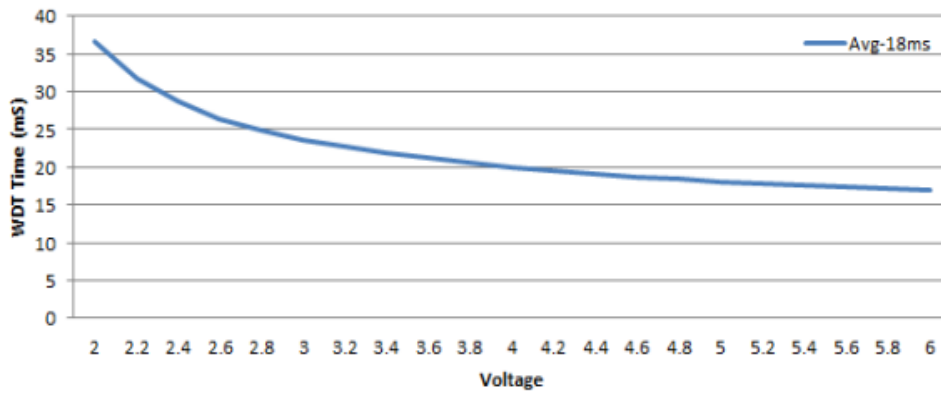
注:曲线仅供设计参考

### 6.2.13 WTD288 毫秒复位时间 vs 温度



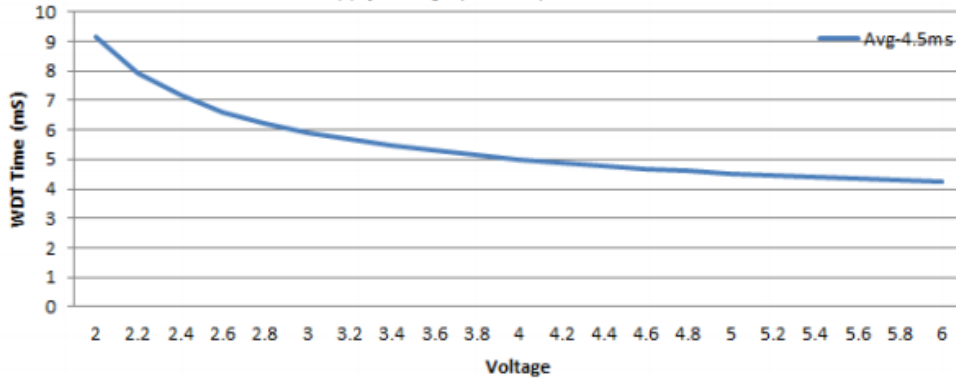
注:曲线仅供设计参考

### 6.2.14 WTD18 毫秒复位时间 vs 供应电压 (Ta=25°C)



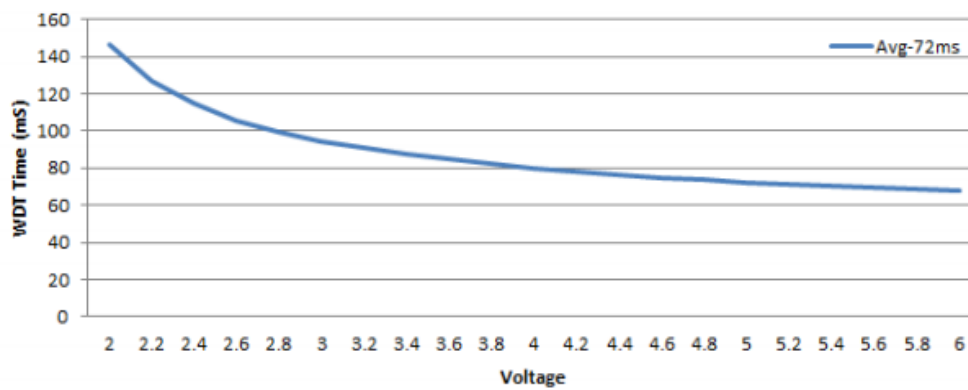
注: 曲线仅供设计参考

### 6.2.15 WTD4.5 毫秒复位时间 vs 供应电压 (Ta=25°C)



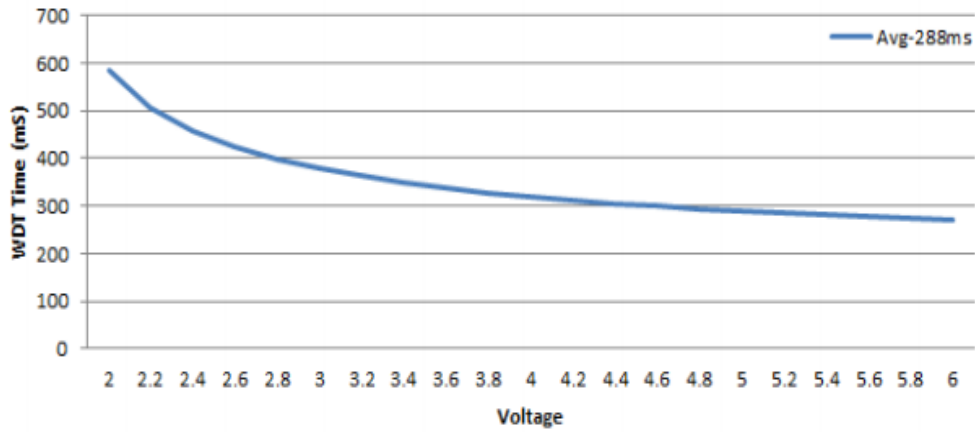
注: 曲线仅供设计参考

### 6.2.16 WTD72 毫秒复位时间 vs 供应电压 (Ta=25°C)



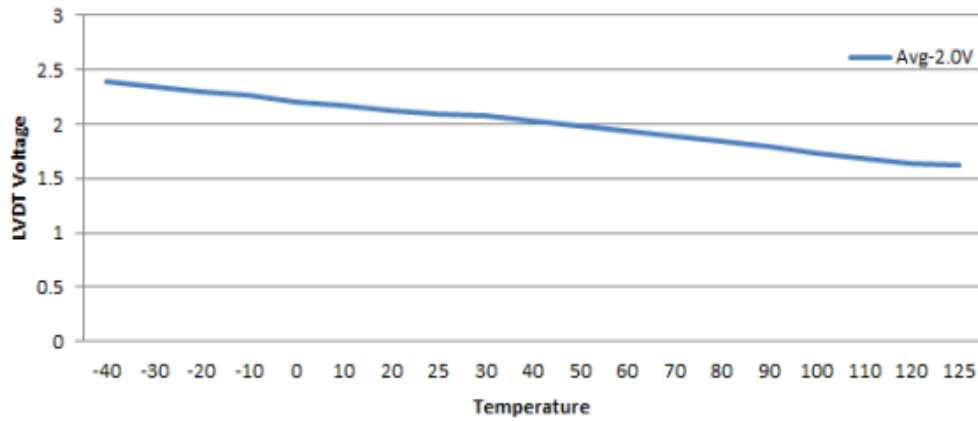
注: 曲线仅供设计参考

6.2.17 WTD288 毫秒复位时间 vs 供应电压 (Ta=25°C)



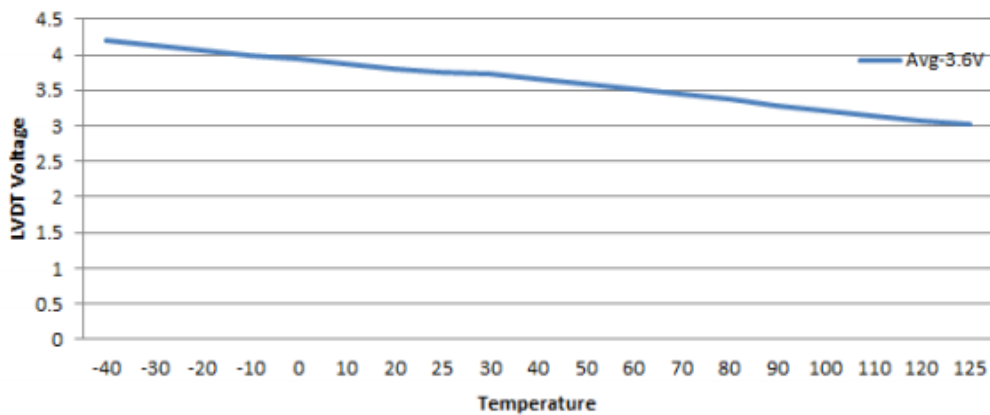
注: 曲线仅供设计参考

6.2.18 LVDT2.0V vs 温度



注: 曲线仅供设计参考

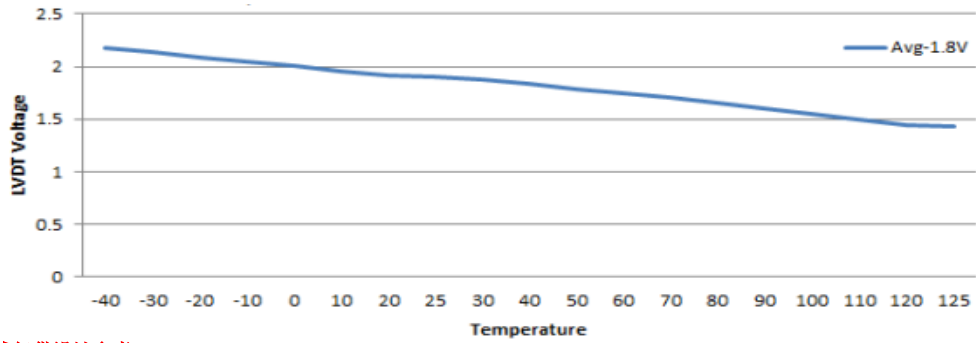
6.2.19 LVDT3.6V vs 温度



注: 曲线仅供设计参考

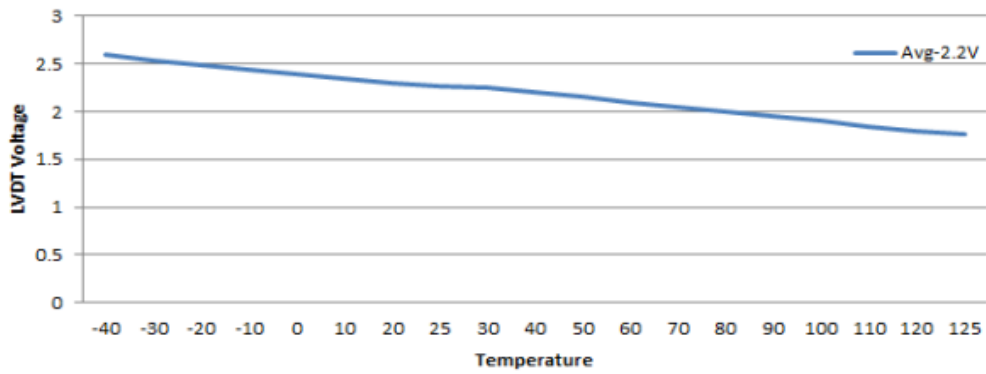


### 6.2.20 LVDT1.8V vs 温度



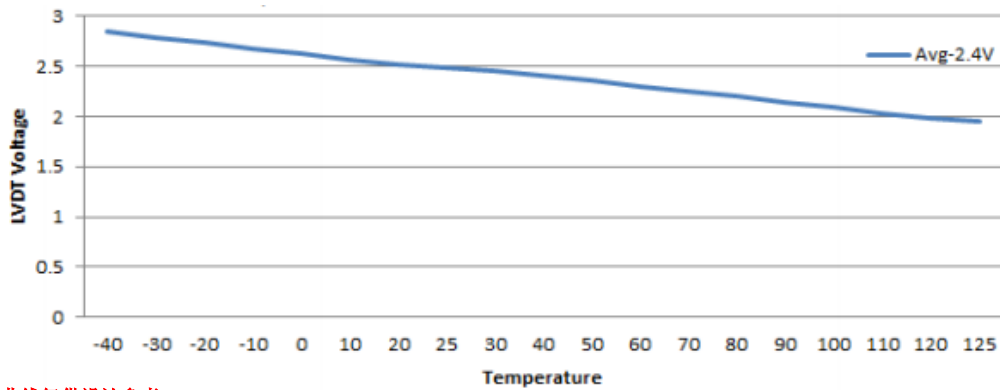
注: 曲线仅供设计参考

### 6.2.21 LVDT2.2V vs 温度



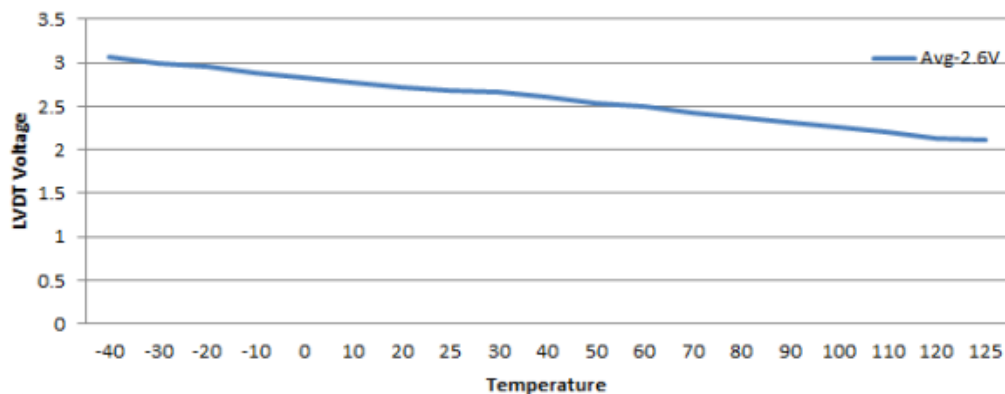
注: 曲线仅供设计参考

### 6.2.22 LVDT2.4V vs 温度



注: 曲线仅供设计参考

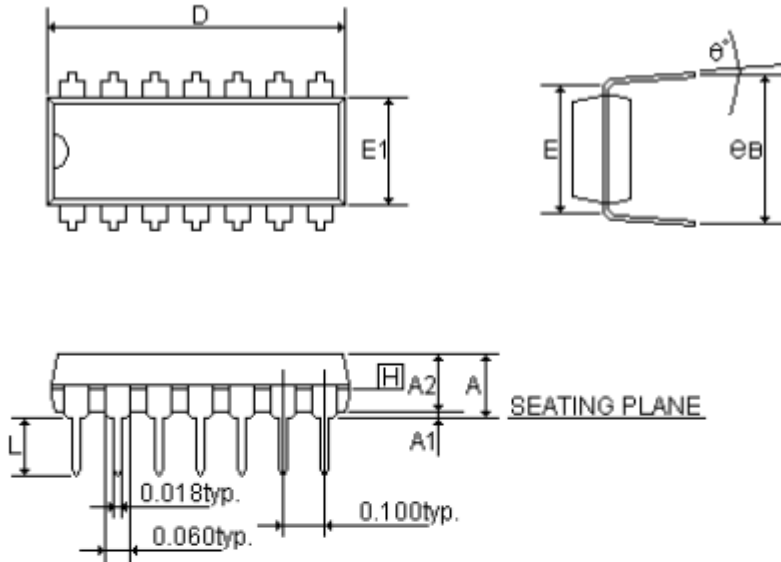
### 6.2.23 LVDT2.6V vs 温度



注: 曲线仅供设计参考

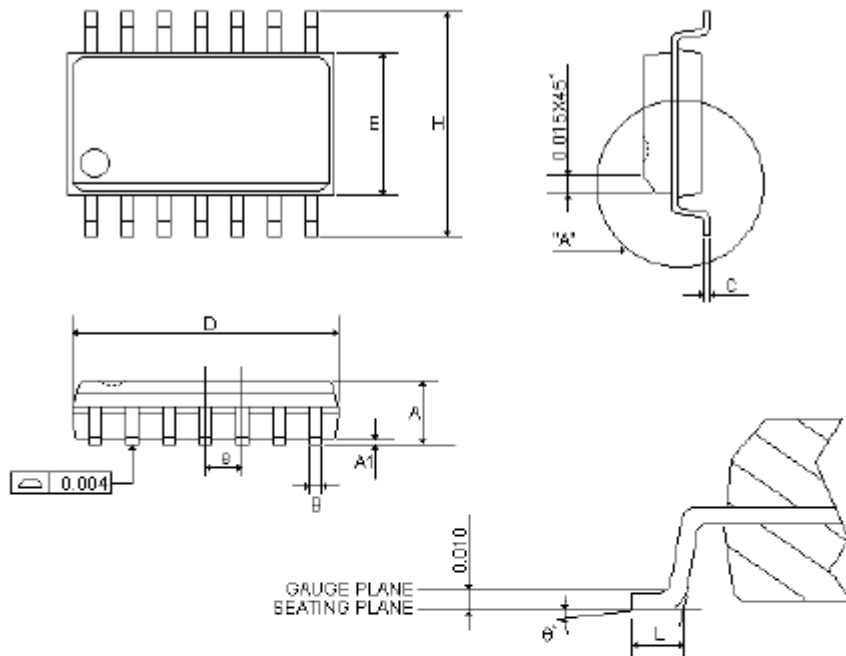
## 7.0 封装尺寸

### 7.1 PDIP 14 PIN



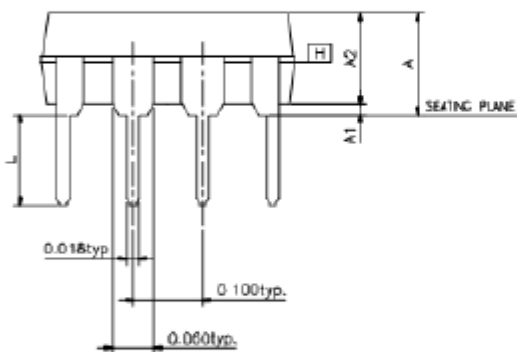
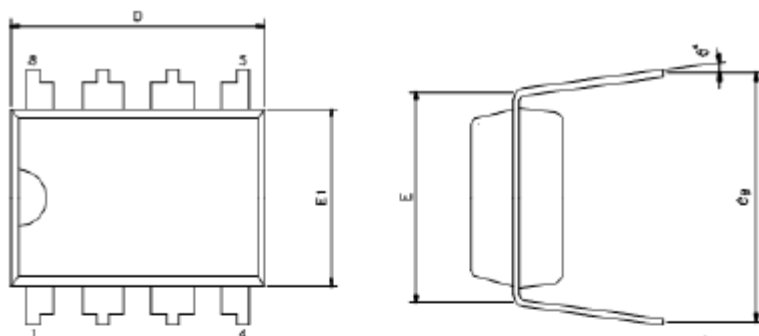
SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
	(inch)			(mm)		
A	-	-	0.210	-	-	5.334
A1	0.015	-	-	0.381	-	-
A2	0.125	0.130	0.135	3.175	3.302	3.429
D	0.735	0.075	0.775	18.669	1.905	19.685
E	0.300			7.62		
E1	0.245	0.250	0.255	6.223	6.35	6.477
L	0.115	0.130	0.150	2.921	3.302	3.810
eB	0.335	0.355	0.375	8.509	9.017	9.525
θ°	0°	7°	15°	0°	7°	15°

## 7.2 SOP 14 PIN



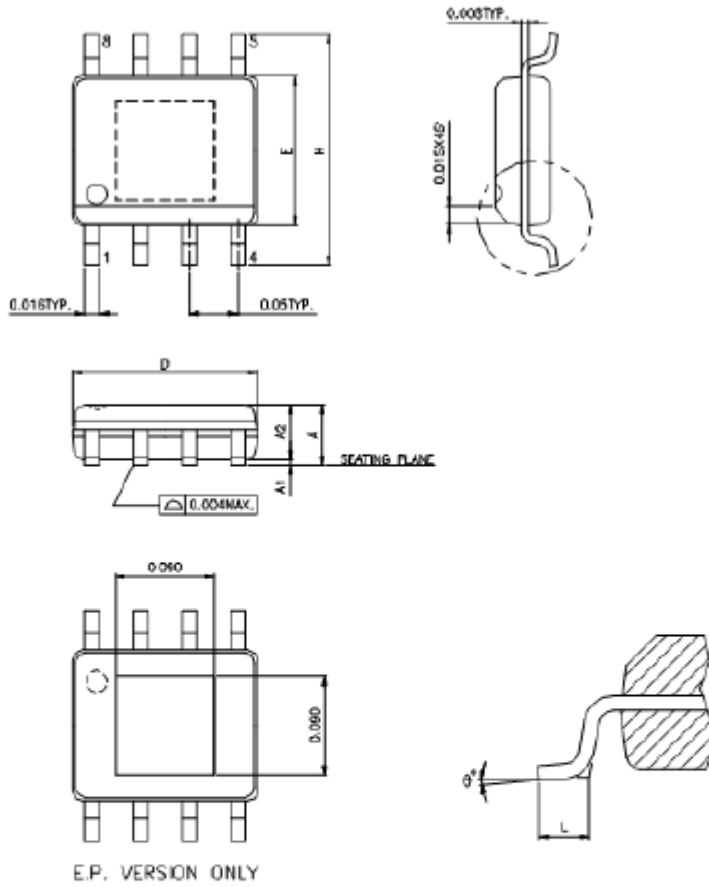
SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
	(inch)			(mm)		
A	0.058	0.064	0.068	1.4732	1.6256	1.7272
A1	0.004	-	0.010	0.1016	-	0.254
B	0.013	0.016	0.020	0.3302	0.4064	0.508
C	0.0075	0.008	0.0098	0.1905	0.2032	0.2490
D	0.336	0.341	0.344	8.5344	8.6614	8.7376
E	0.150	0.154	0.157	3.81	3.9116	3.9878
e	-	0.050	-	-	1.27	-
H	0.228	0.236	0.244	5.7912	5.9944	6.1976
L	0.015	0.025	0.050	0.381	0.635	1.27
$\theta^\circ$	0°	-	8°	0°	-	8°

## 7.3 PDIP 8 PIN



SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
	(inch)			(mm)		
A	-	-	0.210	-	-	5.334
A1	0.015	-	-	0.381	-	-
A2	0.125	0.130	0.135	3.175	3.302	3.429
D	0.355	0.365	0.400	9.017	9.271	10.16
E	0.300			7.62		
E1	0.245	0.250	0.255	6.223	6.35	6.477
L	0.115	0.130	0.150	2.921	3.302	3.810
$\phi B$	0.335	0.355	0.375	8.509	9.017	9.525
$\theta^\circ$	0°	7°	15°	0°	7°	15°

## 7.4 SOP 8 PIN



SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
	(inch)			(mm)		
A	0.053	-	0.069	1.346	-	1.753
A1	0.004	-	0.010	0.102	-	0.254
A2	-	-	0.059	-	-	1.498
D	0.189	-	0.196	4.800	-	4.978
E	0.150	-	0.157	3.810	-	3.988
H	0.228	-	0.244	5.791	-	6.198
L	0.016	-	0.050	0.406	-	1.270
θ°	0°	-	8°	0°	-	8°

## 8.0 包装 IR 回流焊接曲线

